



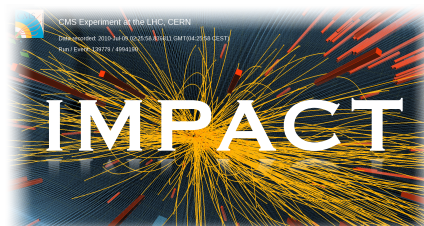
Turning Big data into knowledge

Techniques and Tools for Parallel Computing on Online Data Streams in Systems Biology and Epidemiology

Marco Aldinucci

Computer Science Dept., University of Torino, Italy

Parallel & Distributed Computing group



BioBITs

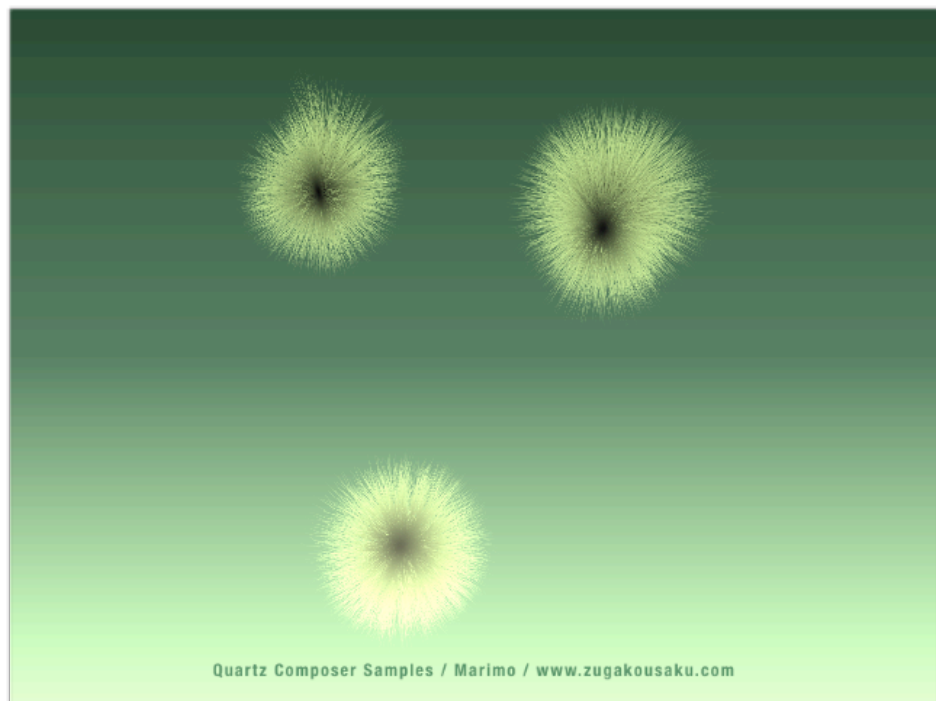
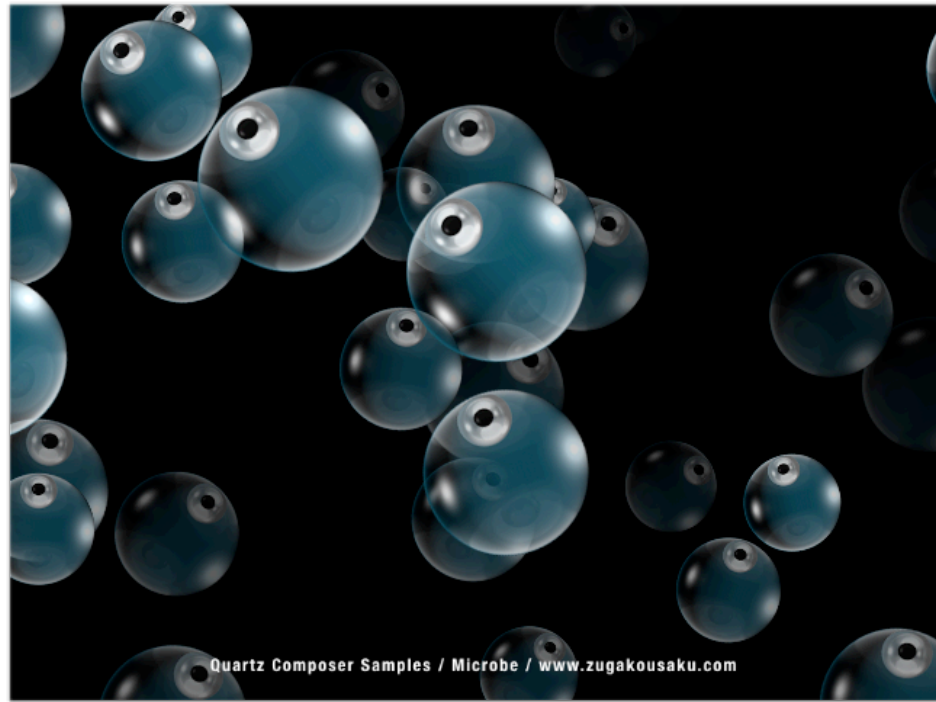


UNIVERSITÀ
DEGLI STUDI
DI TORINO
ALMA UNIVERSITAS
TAURINENSIS

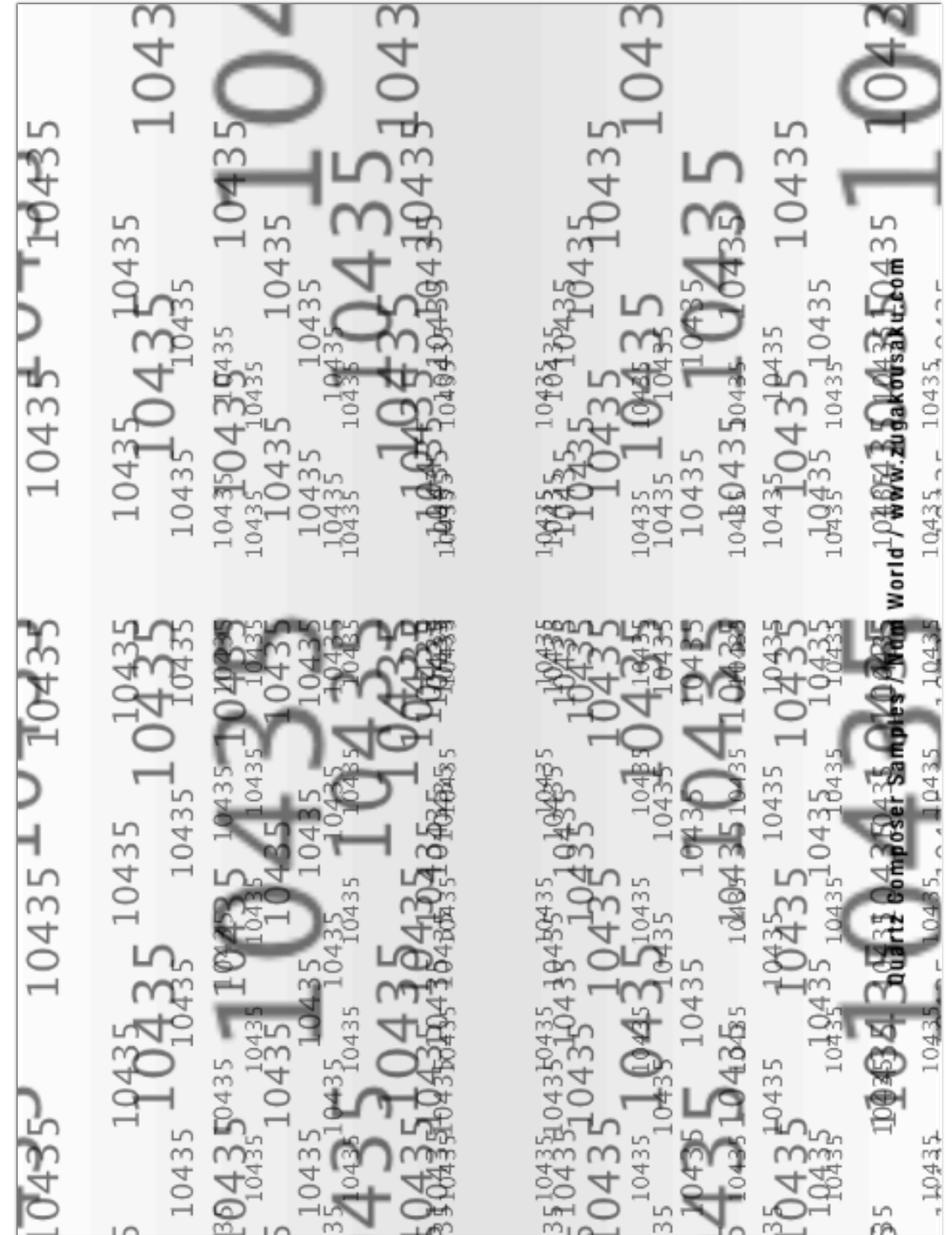


Formal synthetic systems biology

... in my understanding



~



Outline

- Stochastic Formal Systems Biology
 - and produced data
- From Distributed to Multicore and back
- On programming models
 - FastFlow (streaming)
- The CWC parallel simulator for sys bio
- Discussion

Systems Biology & Gillespie's algorithm

- Traditionally studied with continuous ordinary differential equations (ODE)
 - bulk reactions, i.e. average behavior
- Gillespie algorithm: discrete and stochastic simulation of a systems via explicit simulation of each reaction
- Gillespie realization represents a random walk that exactly represents the distribution of the master equation (i.e. ODEs)
 - under some hypothesis

Gillespie's algorithm [77]

1. **Initialization:** Initialize the number of molecules in the system, reactions constants, and random number generators.
2. **Monte Carlo step:** Generate random numbers to determine the next reaction to occur as well as the time interval. The probability of a given reaction to be chosen is proportional to the number of substrate molecules.
3. **Update:** Increase the time step by the randomly generated time in Step 2. Update the molecule count based on the reaction that occurred.
4. **Iterate:** goto Step 2 unless the number of reactants is zero or the simulation time has been exceeded.
5. When many runs are complete, filter and analysis

Increasingly popular approach

- Sometime more informative than (ODE)
 - multi-stability, divergent or rare behaviors, peaks, ...
 - multi-scale systems
 - e.g. deriving macro behavior from micro
- More computational demanding
 - much more, especially in motivating cases

Increasingly popular approach

- Bio-PEPA [Hillston, Ciocchetta]
- SPiM [Cardelli, Phillips]
- Stochastic Pi [Priami]
- Stochkit [Petzold]
- Spatial Pi [Uhrmacher]
- Calculus of Wrapped Components [our own]
 - kinetics: mass-action, Michaelis–Menten, Hill ...
- ...

Formalising the cell cycle switch

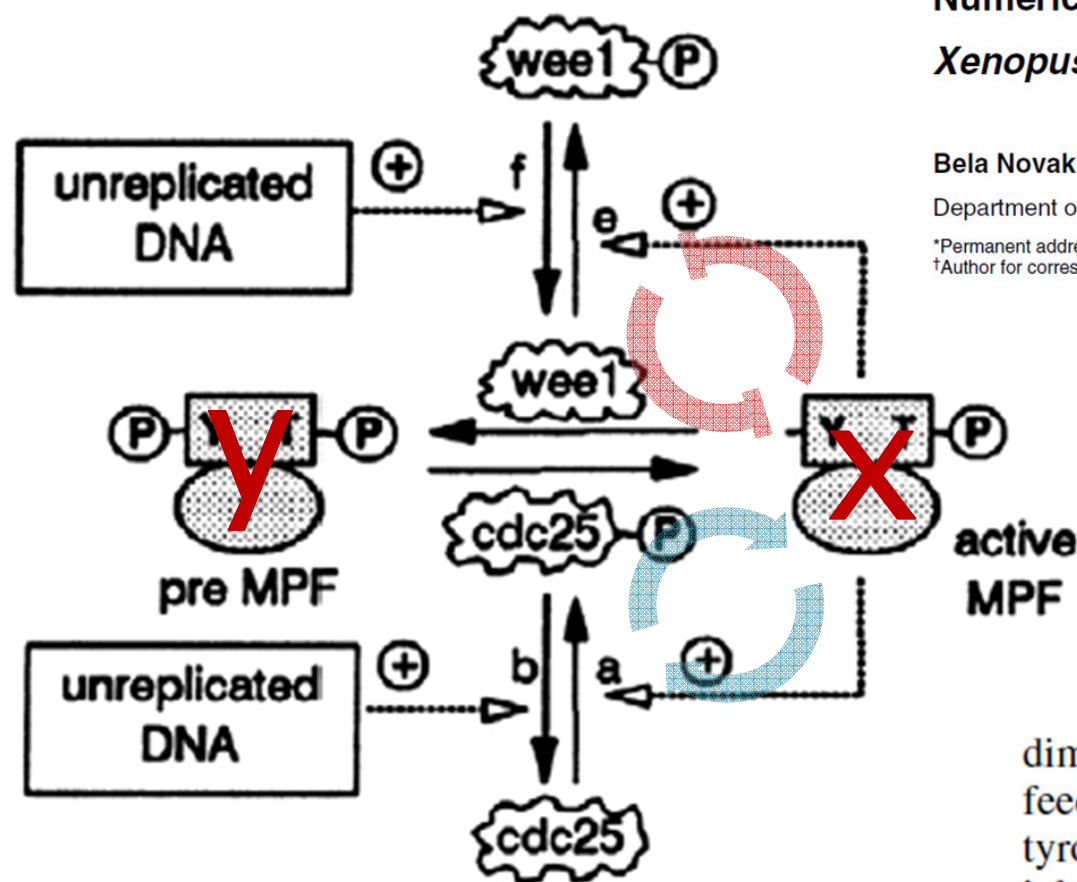
Journal of Cell Science 106, 1153-1168 (1993)
Printed in Great Britain © The Company of Biologists Limited 1993

Numerical analysis of a comprehensive model of M-phase control in *Xenopus* oocyte extracts and intact embryos

Bela Novak* and John J. Tyson†

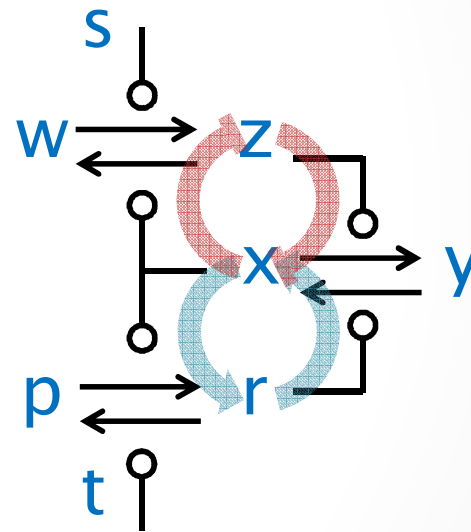
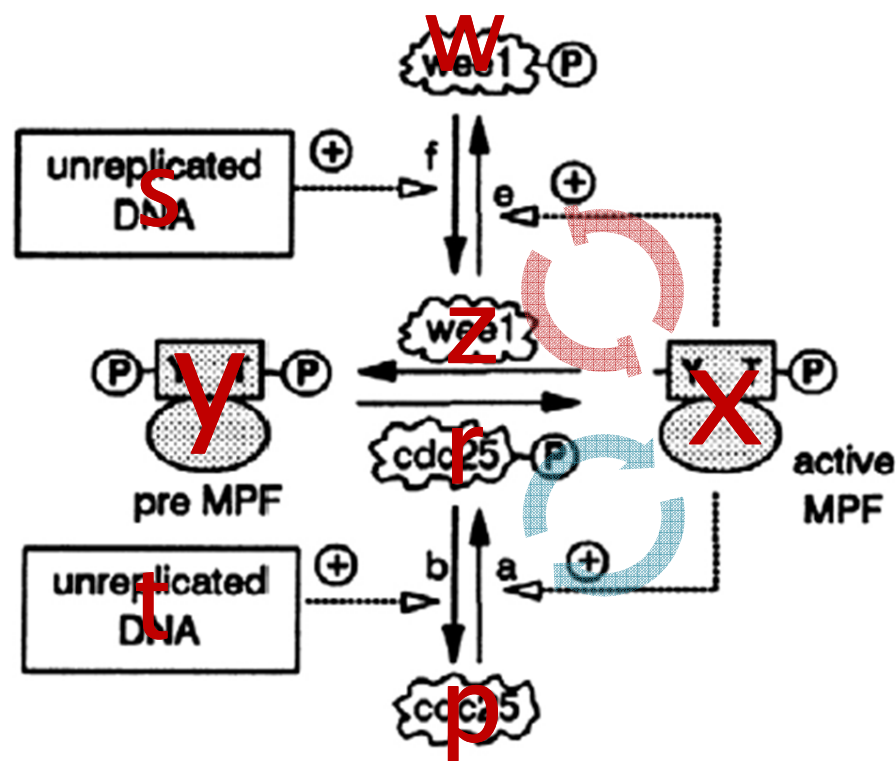
Department of Biology, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24060-0406, USA

*Permanent address: Department of Agricultural Chemical Technology, Technical University of Budapest, 1521 Budapest Gellert Ter 4, Hungary
†Author for correspondence



dimers is left off the diagram to keep it simple.) (B) Positive feedback loops. Active MPF stimulates its own production from tyrosine-phosphorylated dimers by activating Cdc25 and inhibiting Wee1. We suspect that these signals are indirect, but intermediary enzymes are unknown and we ignore them in this paper. The signals from active MPF to Wee1 and Cdc25 generate an autocatalytic instability in the control system. We indicate also an 'external' signal from unreplicated DNA to Wee1 and Cdc25, which can be used to control the efficacy of the positive feedback loops. The letters a, b, e and f are used to label the rate constants for these reactions in Fig. 2. (C) Negative feedback loop. Active

... a switch, when formalised



Courtesy of Luca Cardelli
 On Switches and Oscillators Program
 Equivalence in Biology?
<http://lucacardelli.name>

$$\top : a \ c \xrightarrow{10} c \ b$$

$$\top : c \ a \xrightarrow{10} a \ b$$

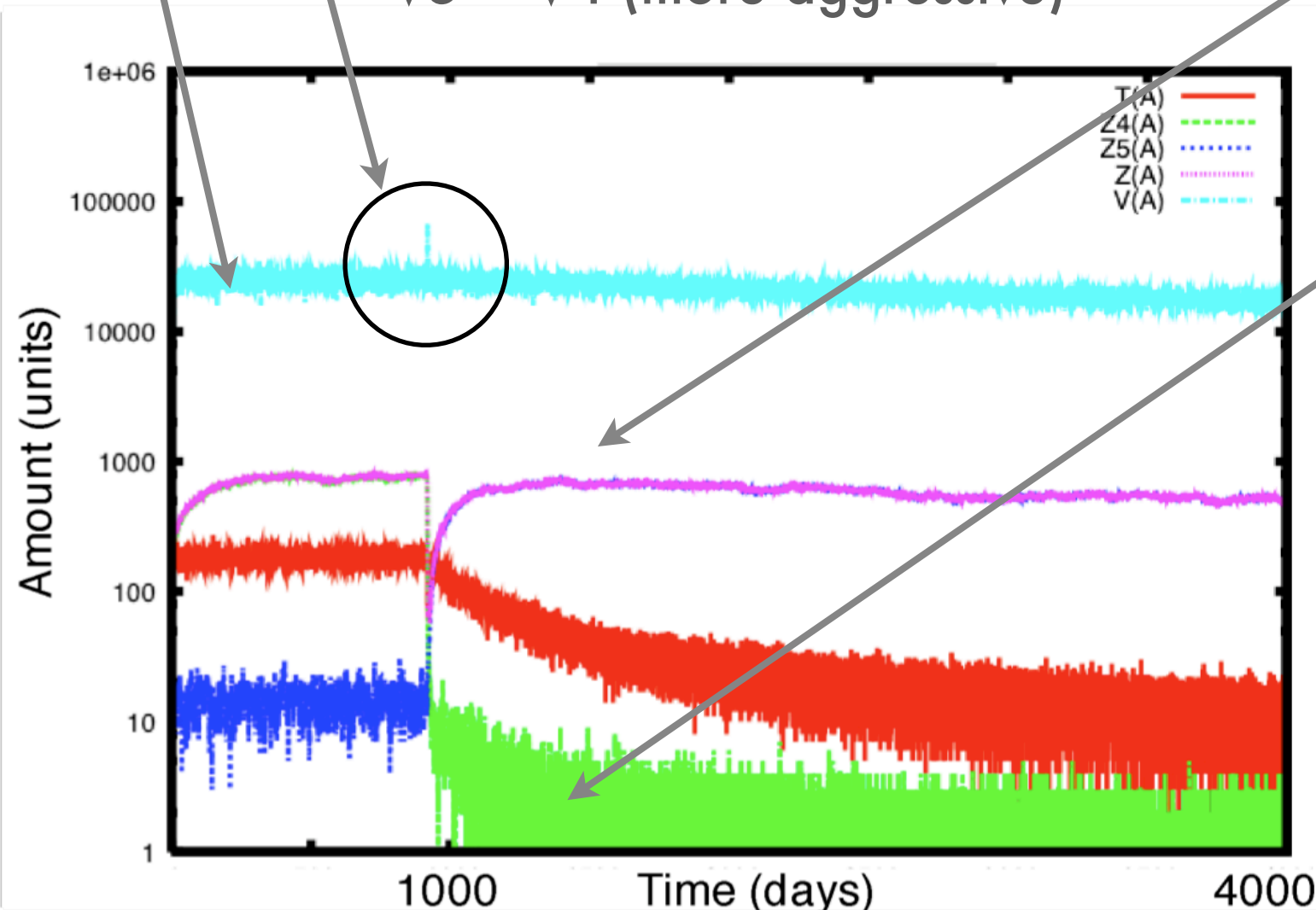
$$\top : b \ a \xrightarrow{10} a \ a$$

$$\top : b \ c \xrightarrow{10} c \ c$$

Ex: HIV and immune response (progression to AIDS)

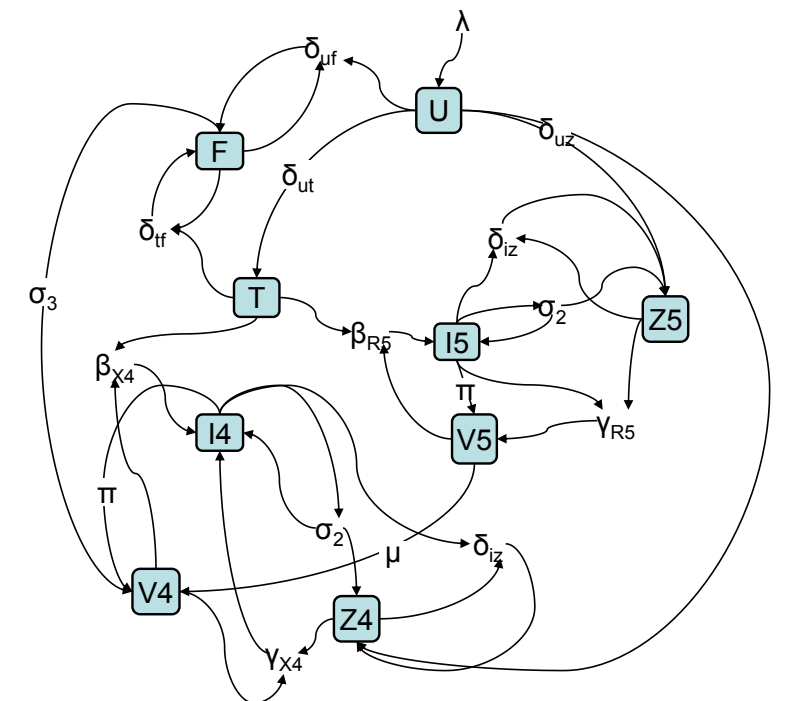
V virus, all phenotypes (V4,V5)

the spike suggests the mutation
V5 $\rightarrow$ V4 (more aggressive)



immune response $Z = Z_4 + Z_5$
remain stable (but for the peak).

now **Z4** decrease and
Z5 increase
i.e. HIV is turning into
AIDS more rapidly



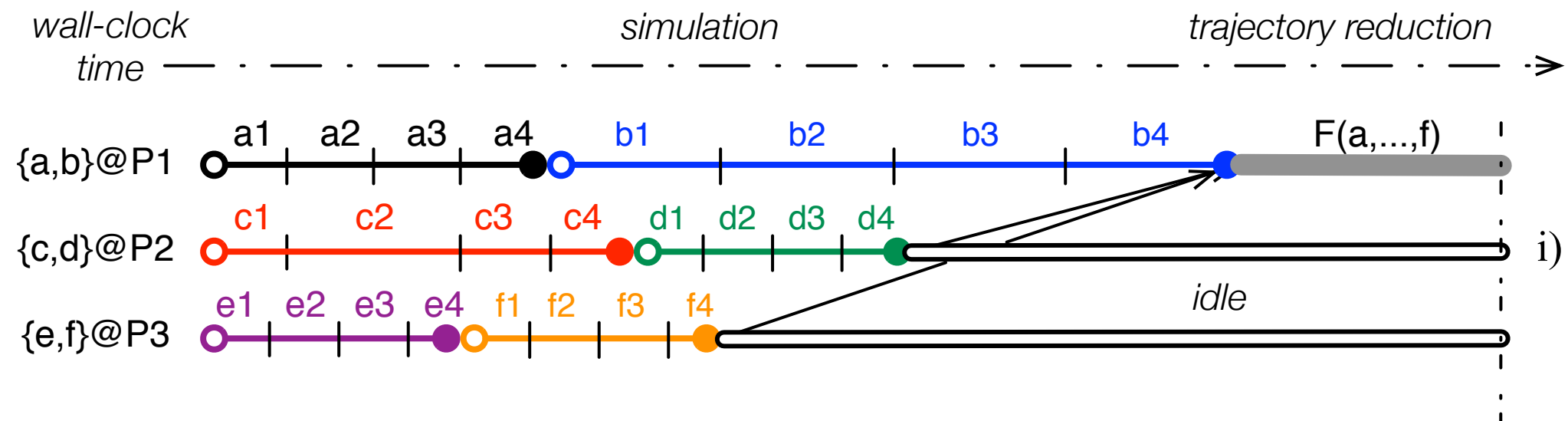
- Peaks are informative events,
 - virus mutation triggers AIDS progression
 - hardly detected with ODEs
- high resolution required to detect spikes,
 - each trajectory can be over 6G Bytes of data
 - The more observed variables, resolution/precision, cases for sensitivity analysis, the more data
- and thousands of trajectories are needed
 - compute everything, save everything, move and join all data, analyse all data, then get first results
 - often to discover parameters are wrong ...



- It is Monte Carlo,
 - well understood
 - easy to parallelise on different trajectories
- it is Monte Carlo w Markov Chains models (CTMC)
 - single trajectory: no parallelisation without relaxation
 - compute time $\neq$ simulation time
 - compute time for different trajectories heavily unbalanced
 - fast reactions and slow reactions, some not interesting (e.g. water-steam-water)



Unbalancing + filtering

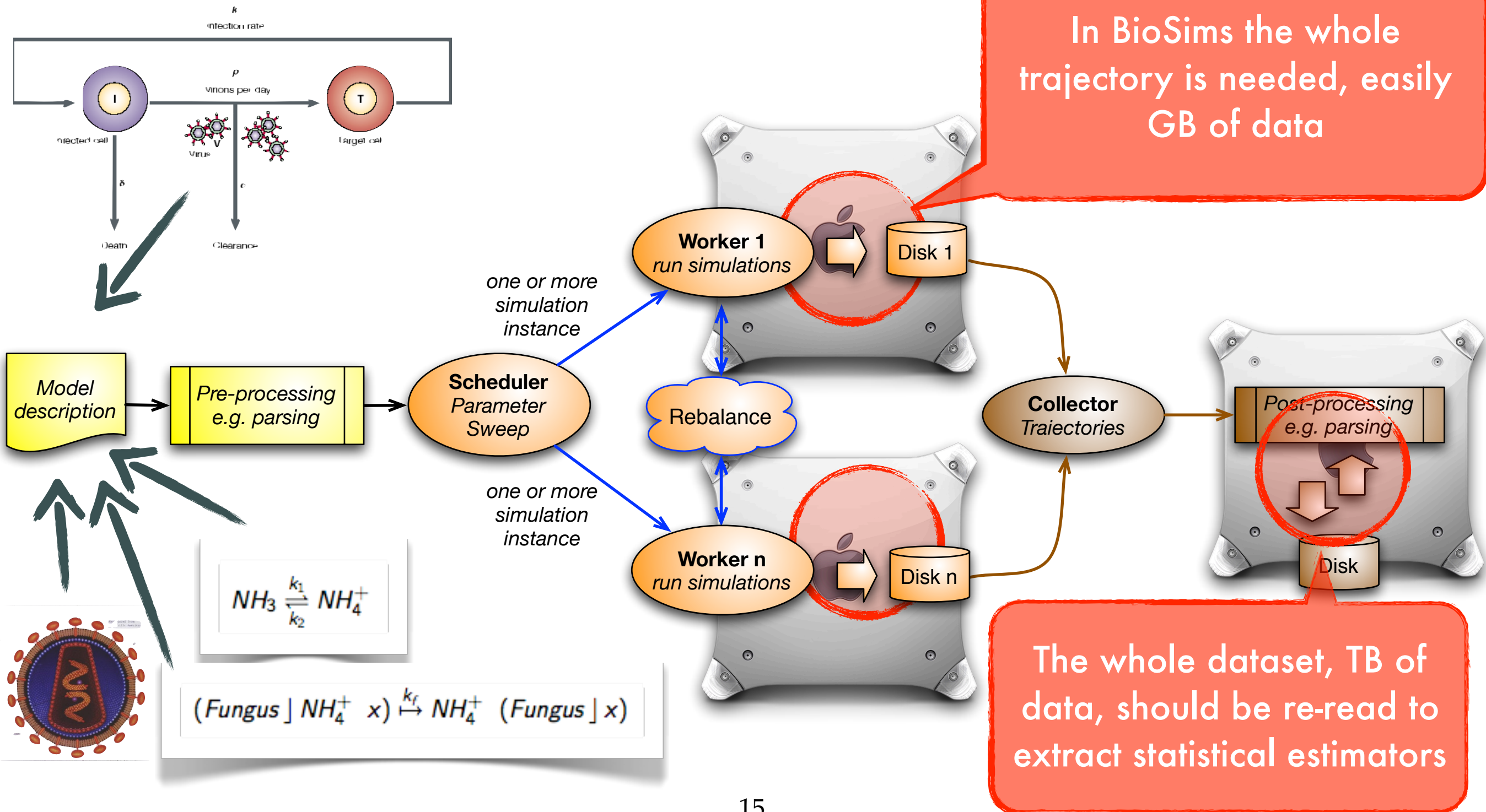


- few trajectories (e.g. the interesting ones) can significantly delay the completion of others
 - over-provisioning don't help that much
 - simulated time moves at different pace w.r.t. wall-clock time
 - data joining from different trajectories should be aligned at the same simulation time

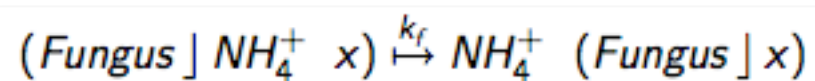
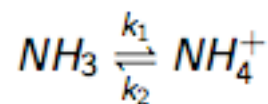
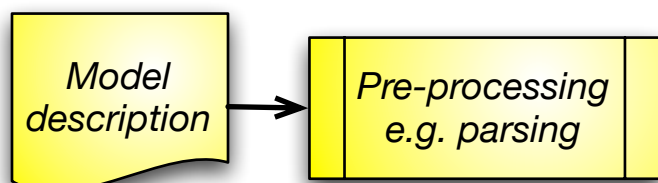
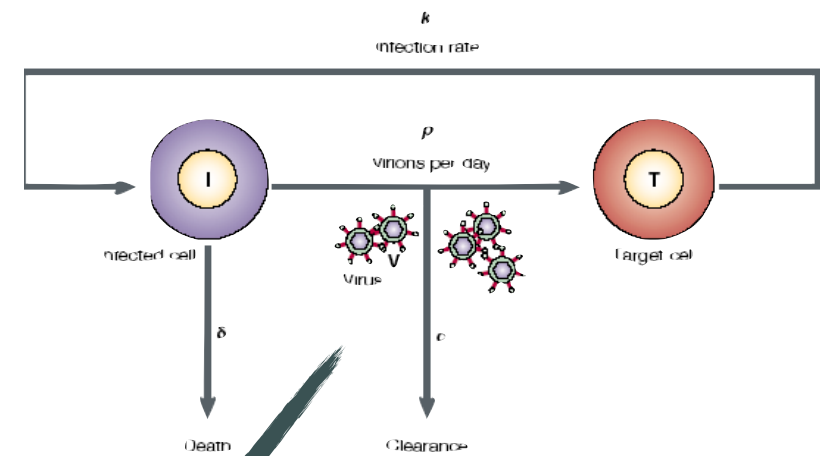
- It is Monte Carlo,
 - well understood
 - easy to parallelise on different trajectories
- it is Monte Carlo w Markov Chains models (CTMC)
 - single trajectory: no parallelisation without relaxation
 - compute time $\neq$ simulation time
 - compute time for different trajectories heavily unbalanced
 - fast reactions and slow reactions, some not interesting (e.g. water-steam-water)
- it is Monte Carlo AND data analysis
 - data is big, analysis can be very expensive and **it typically starts after the simulation**
 - the whole workflow is perceived too “slow” by bio-scientists to be really useful



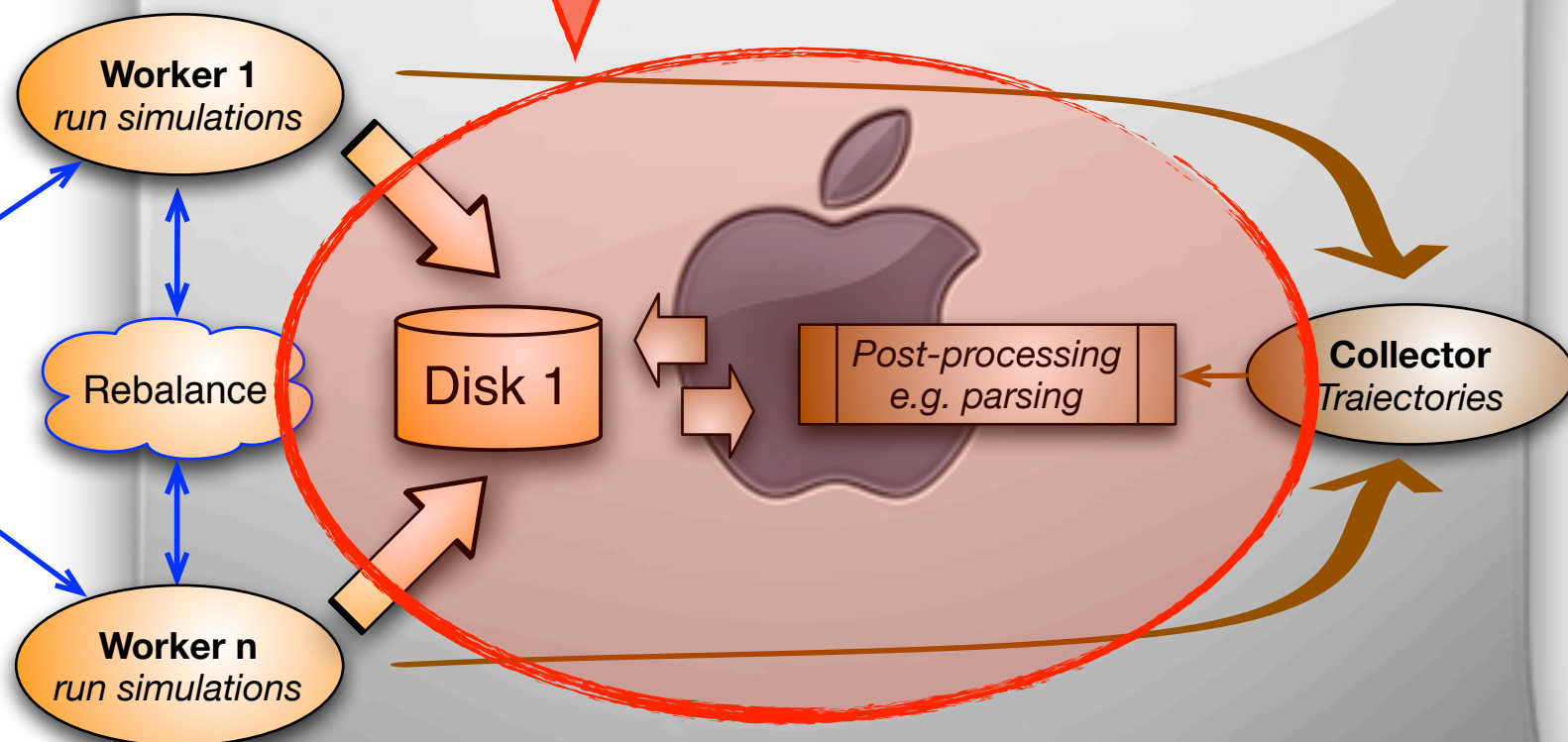
From Distributed to Multicore and back



From Distributed to Multicore and back



Now the issue become a real problem
Bottlenecks: disk and memory
Post-processing: not pipelined and often not parallel at all



From Distributed to Multicore and back

- Multi Carlo sims for Bio are I/O-bound
 - Sampling reduce I/O traffic but worsen precision and analysis of “strange” dynamics (spikes, diversion from average, etc.), which observation motivates stochastic analysis (ODEs)
- Data analysis is also I/O-bound
 - if approached is a “post-processing” fashion, data should be retrieved from the disks
- The porting of distributed solution “as is” on multicore is going insist on weakness of multicore architectures
 - Memory wall, I/O, disk
 - SIMD / GPGPUs do not change the analysis substantially

From Distributed to Multicore and back

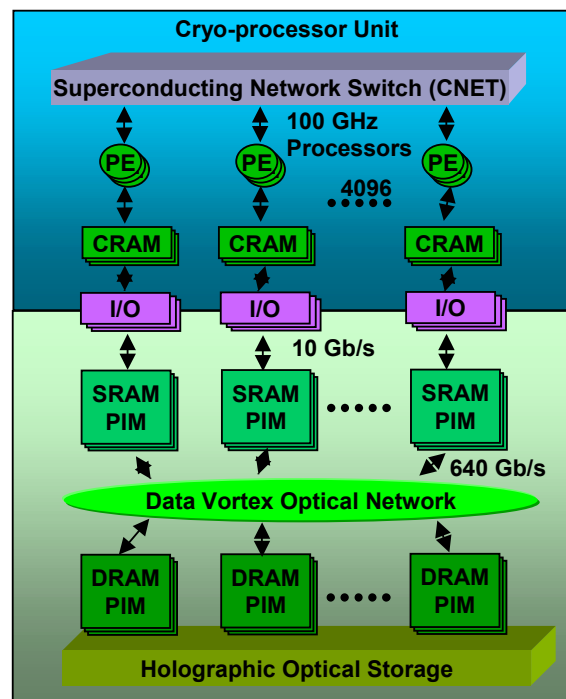
- The same arguments holds on distributed, grids, and clouds as soon as the workflow is considered as a whole
 - simulation, data collection and merging, analysis
- Rationale
 - **Manage data as stream, compute online**
 - May require more computing and less bandwidth
 - Computation should be designed to be pipelined
 - **Establish fast data paths across cores/hosts**
 - **Avoid low-level concurrency management**
 - Portability, performance, portability of performance, maintenance, porting from sequential

Reasoning on next generation (is a slippery ground)

PetaFlops

(Santa Barbara, USA, Feb 1998)

Hybrid Technology Multi-Threaded Computer



PETAFLIPS II
BELLEVUE II

February 15-19, 1999 Doubletree Hotel Santa Barbara, CA

- **Hybrid Technology**

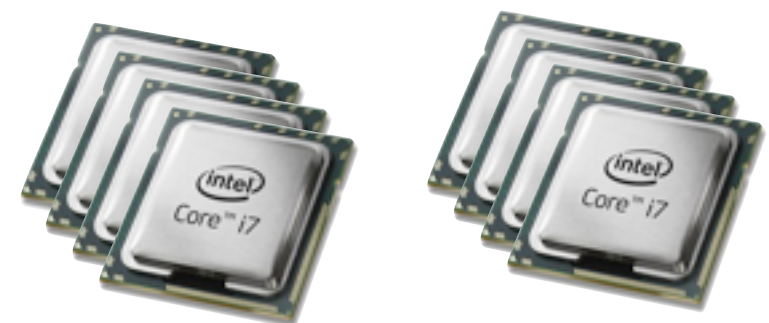
- 100 GHz SCE Processors (RSFQ)
- Processor in Memory (PIM) SRAM/DRAM
- Terabit optical network (WDM)
- Holographic memory (Spectral hole burning)

- **Multi-Thread**

- multi-stage latency management
- memory "percolation"
- Coarse grain multithreading
- PIM smart memory management

Impact	TODAY	HTMT	IMPROVE
PERFORMANCE	1 Teraflops	1000 Teraflops	X 1000
POWER	2 Mflops/W	1000 Mflops/W	X 500
COST	\$250/Mflops	\$0.25/Mflops	X 1000
EFFICIENCY	10%	50%	X 5
FLOORSPACE	1600 sq ft/Tflop	1 sq ft/Tflop	X 1600

Today



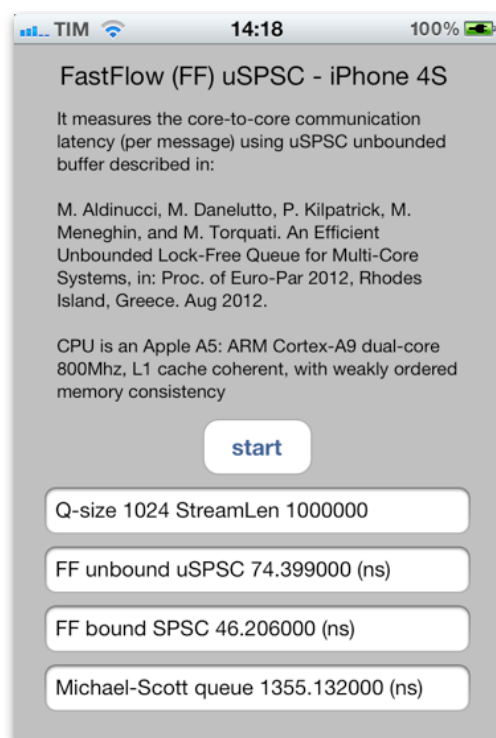
Exa-scale?

- *programming systems should be designed to support fast data movement and enforce locality*
 - *coarse grain concurrency is nearly exhausted*
 - *it is not about Flops, it is about data movement*
 - *shared-memory & inter-socket messaging*
- *we need a **programming model***
 - *a computer language is not a computing model!*
a library is not a computing model!
- *we need a **efficient** and **compositional** run-time*



<http://mc-fastflow.sourceforge.net/>

FastFlow



- C++ pattern-based framework, open-source LGPL
- A tiny, lightweight & open research framework for HPC
 - 5K lines of code
- 3 years old - over 8K downloads - over 40K contacts
- x86/PPC/ARM + Linux/Mac/Win/iOS
- Multicore, GPGPU, distributed (TCP & Infiniband)

FastFlow (multicore)

Applications on multicore, many-core
Efficient and portable - designed with high-level patterns

FastFlow

Streaming network patterns
Skeletons: pipeline, map farm, reduce, D&C, ...

Arbitrary streaming networks
Lock-free SPSC/MPMC queues + FF nodes

Simple streaming networks
Lock-free SPSC queues + threading model

Multicore and manycore
SMP: cc-UMA & cc-NUMA

Layer 1: Simple streaming networks

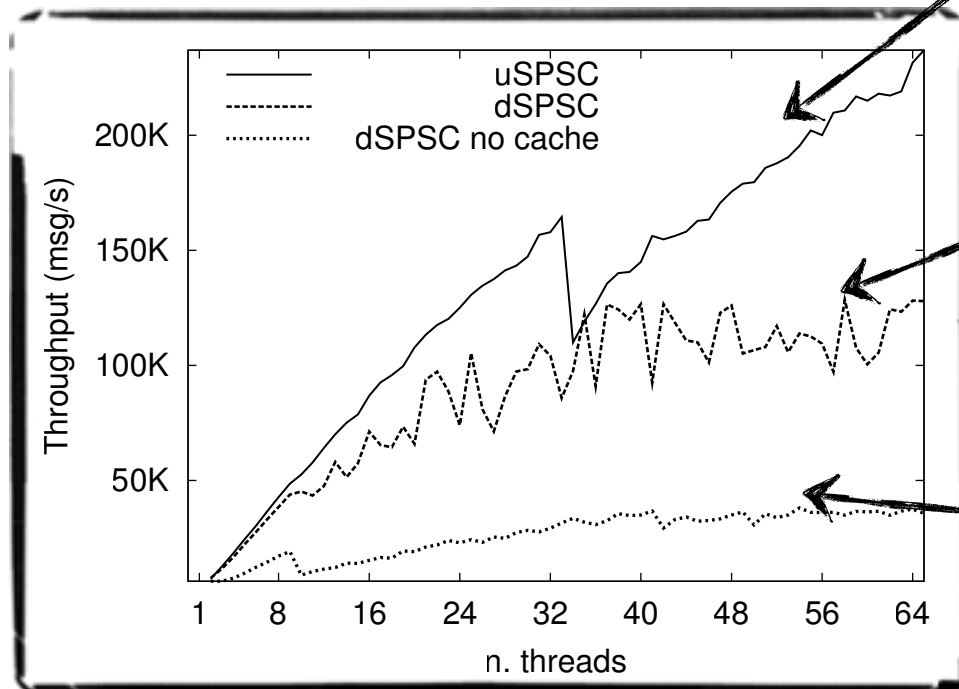
<http://www.1024cores.net/home/technologies/fastflow>

4 sockets x 8 core x 2 contexts

Xeon E7-4820 @2.0GHz Sandy Bridge
18MB L3 shared cache, 256K L2

8ns per message (same socket)
35ns per message (diff sockets)
as reference: MPI-shmem 190ns

Speedup



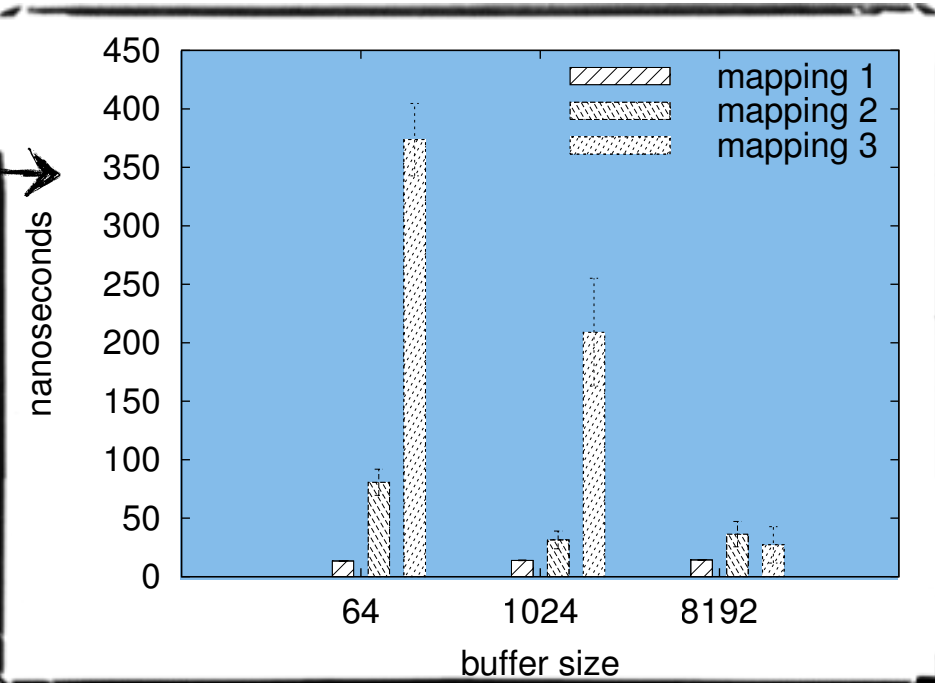
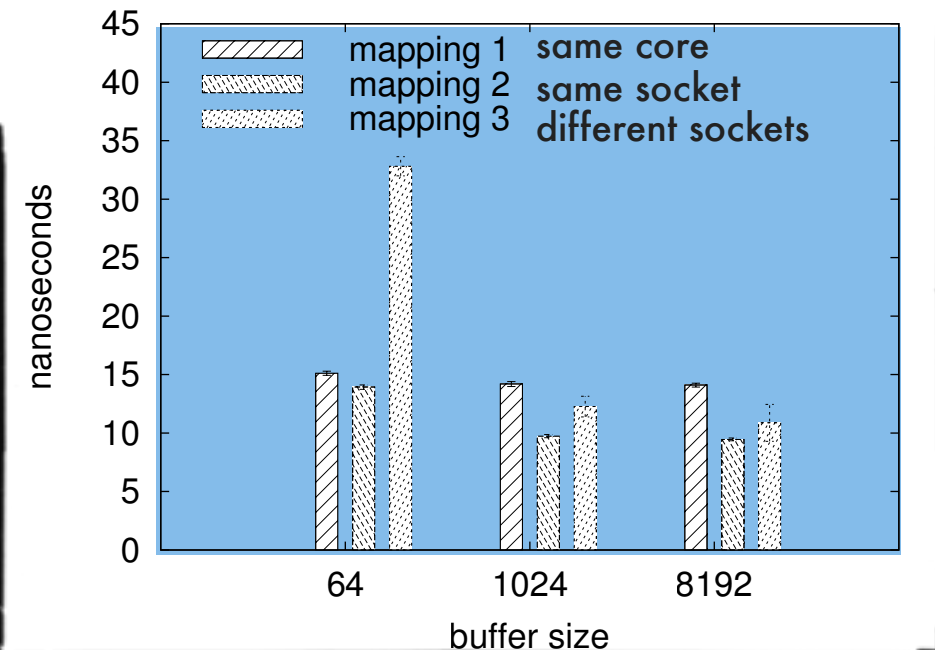
Linked list + circular buffer
FastFlow queue
(our result)

12x faster

Linked list with pooling
Opt Michael-Scott queue
(our result)

20x faster

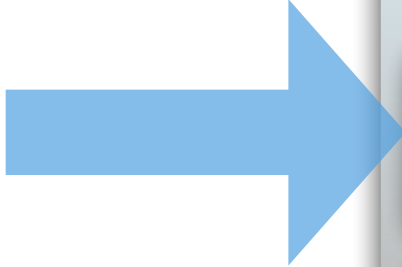
Linked list w/ dyn alloc
Michael-Scott queue
well-known ~ 400 citations



FastFlow (multicore)

Applications on multicore, many-core
Efficient and portable - designed with high-level patterns

FastFlow



Streaming network patterns
Skeletons: pipeline, map farm, reduce, D&C, ...

Arbitrary streaming networks
Lock-free SPSC/MPMC queues + FF nodes

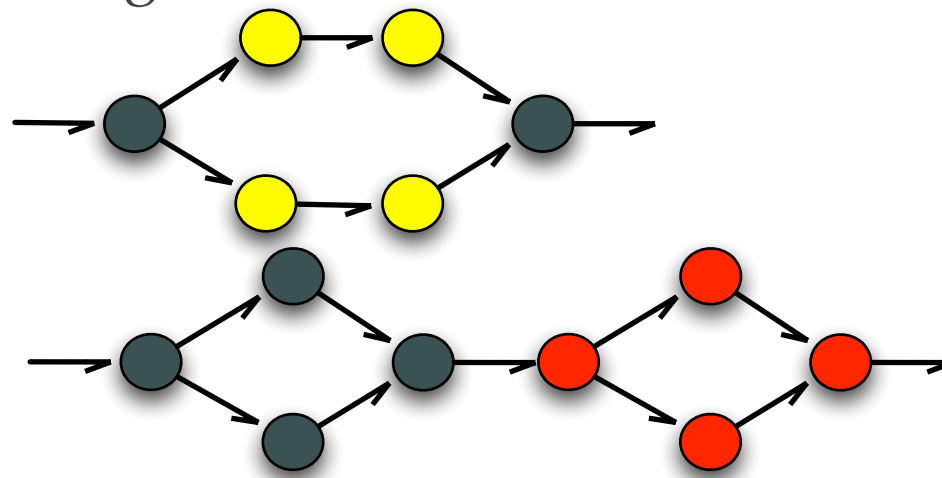
Simple streaming networks
Lock-free SPSC queues + threading model

Multicore and manycore
SMP: cc-UMA & cc-NUMA

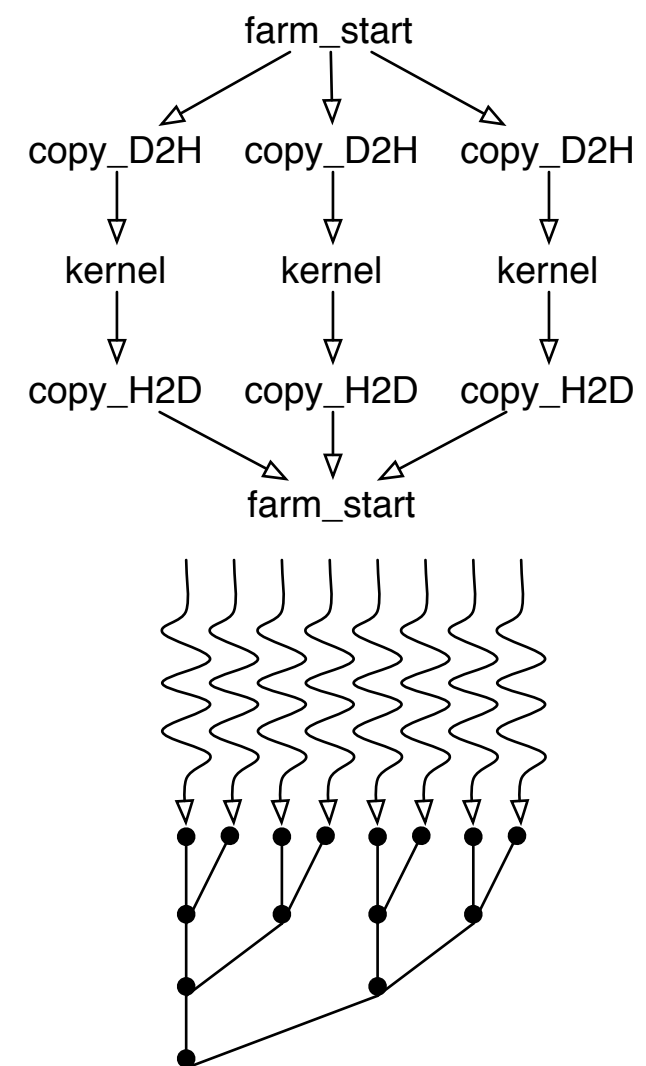
Layer 3: streaming networks patterns

- Composition via C++ template meta-programming
 - CPU: Graph composition
 - GPU: CUDA streams
 - CPU+GPU: offloading
- `farm{ pipe }`
- `pipe(farm, farm)`
- `pipe(map, reduce)`
-

Multi-core



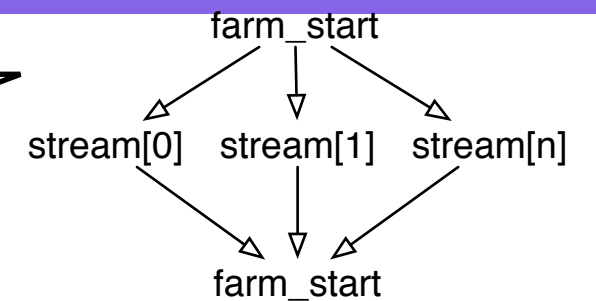
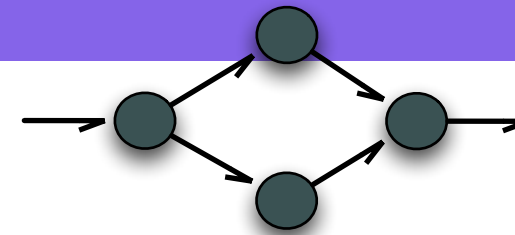
GPGPU



Layer 3: streaming networks patterns

- farm

- on CPU - master-worker - parallelism exploitation
- on GPU - CUDA streams - automatic exploitation of asynch comm



- pipeline

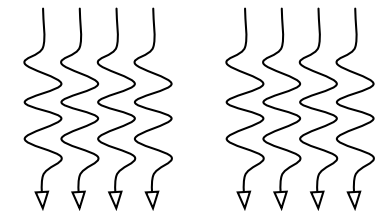
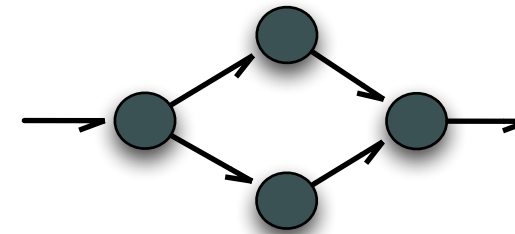
- on CPU - pipeline
- on GPU - sequence of kernel calls or global mem synch



stream[k]
 copy_D2H
 ∇
 kernel
 ∇
 copy_H2D

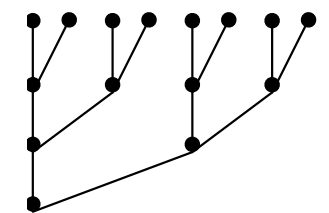
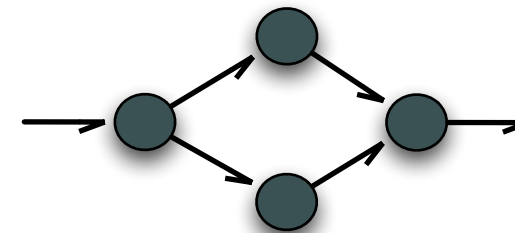
- map

- on CPU - master-worker - parallelism exploitation
- on GPU - CUDA SIMT - parallelism exploitation



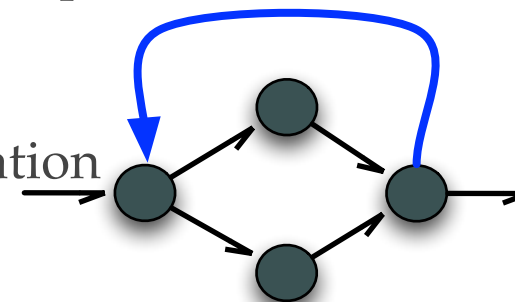
- reduce

- on CPU - master-worker - parallelism exploitation
- on GPU - CUDA SIMT (reduction tree) - parallelism exploitation



- D&C

- on CPU - master-worker with feedback - // exploitation
- on GPU - working on it, maybe loop+farm



+ distributed

Applications on multicore, many core & distributed platforms of multicores
Efficient and portable - designed with high-level patterns

FastFlow

Streaming network patterns

Skeletons: pipeline, map farm, reduce, D&C, ...

Arbitrary streaming networks

Lock-free SPSC/MPMC queues + FF nodes

Arbitrary streaming networks

Collective communications + FF Dnodes

Simple streaming networks

Lock-free SPSC queues + threading model

Simple streaming networks

Zero copy networking + processes model

Multicore and manycore

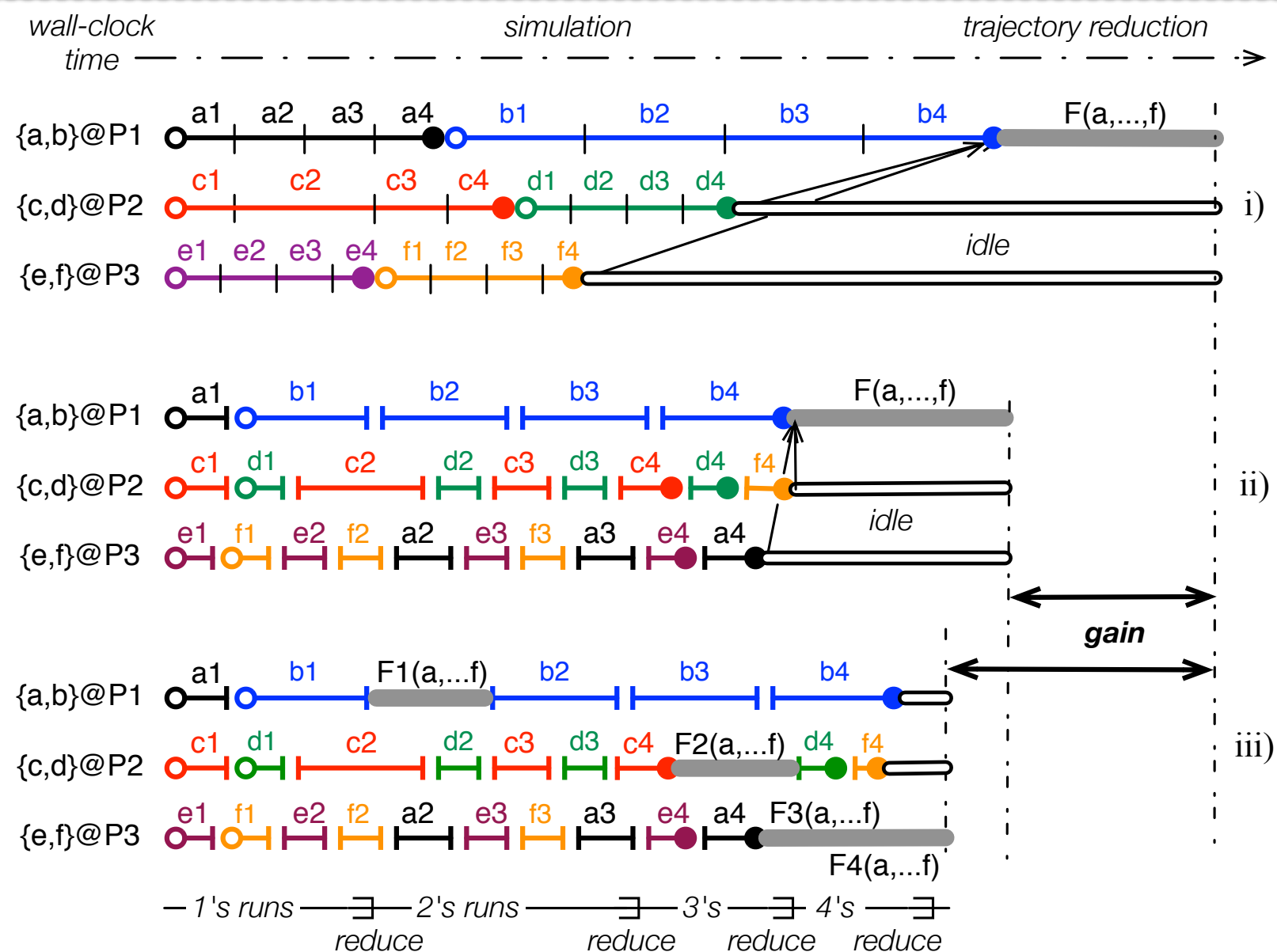
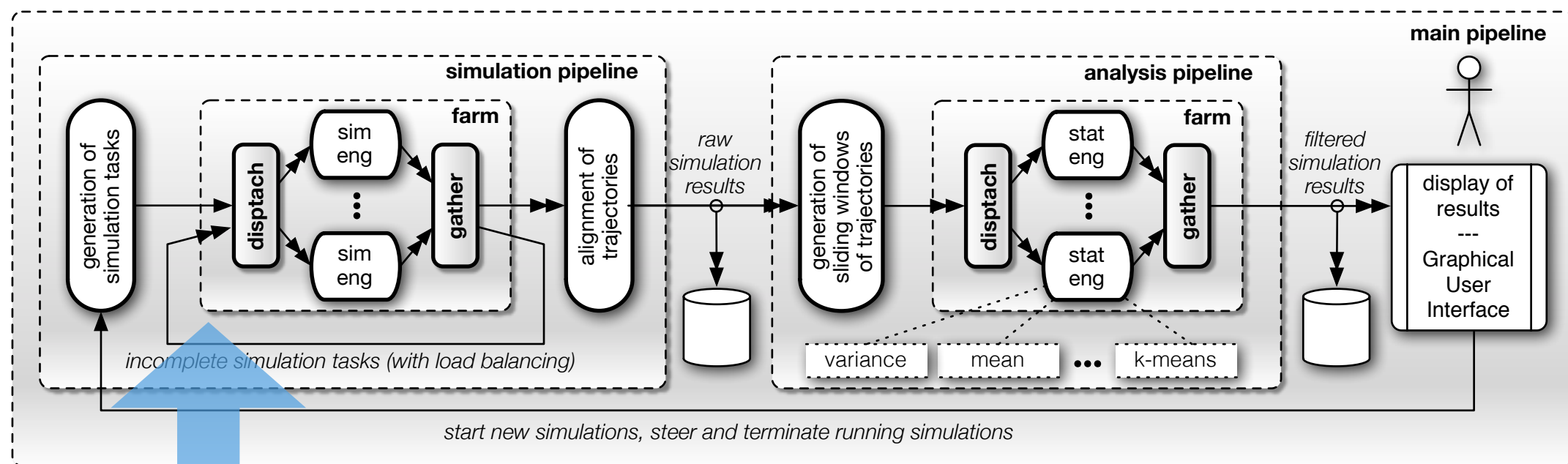
SMP: cc-UMA & cc-NUMA

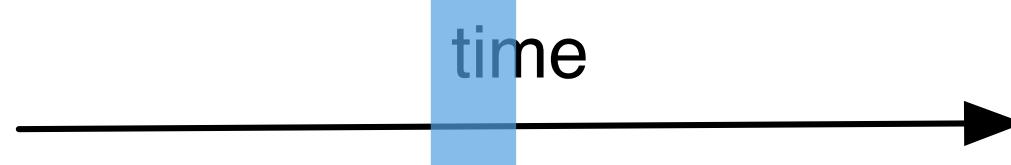
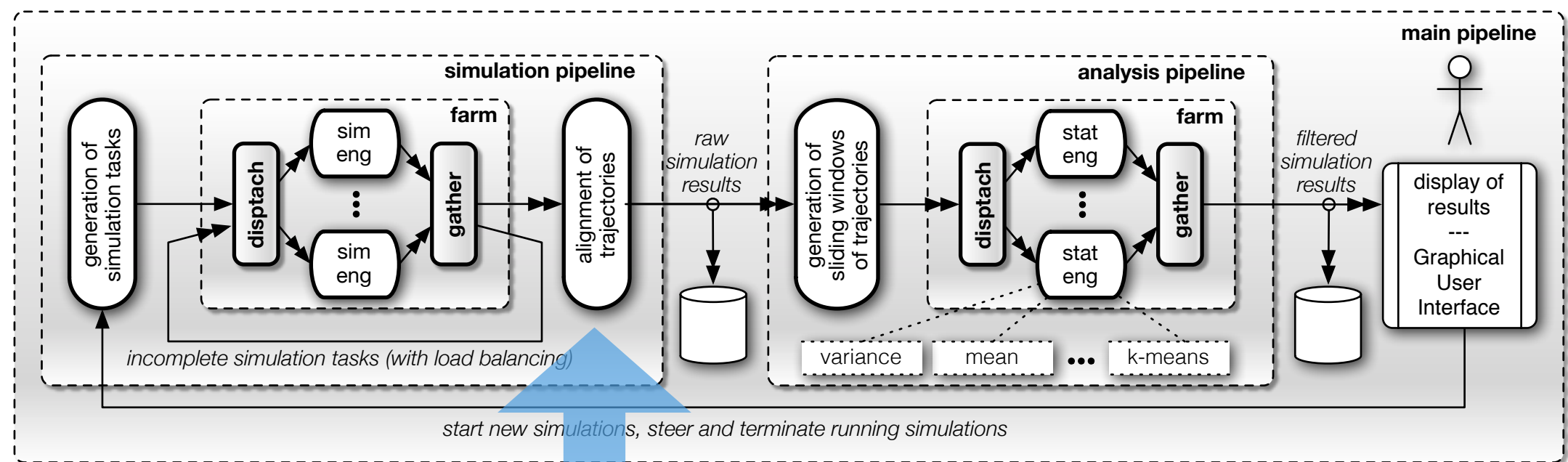
Distributed platforms

Clouds, clusters of SMPs

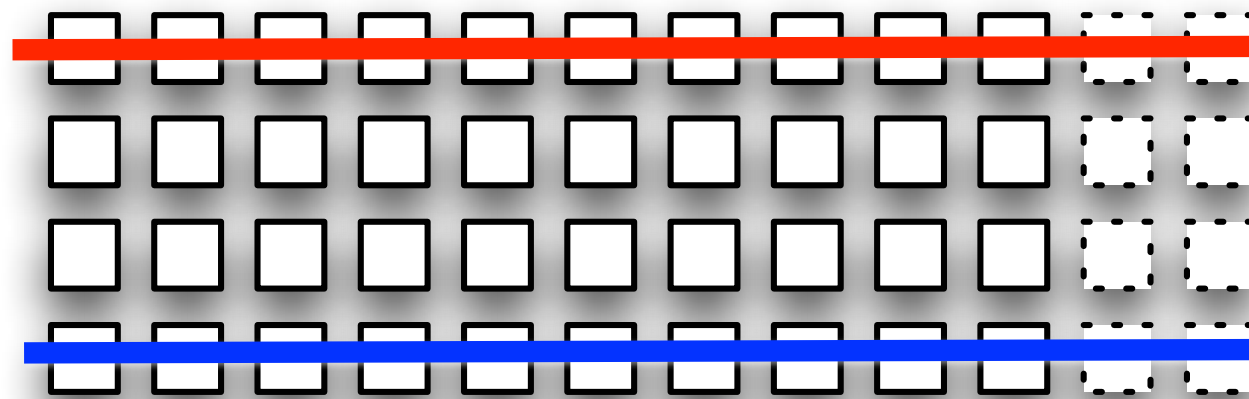
- network channels
 - P2P or collective
 - used as frontier node of streaming graph
 - can be used to merge graphs across distributed platforms
- No changes to programming model
 - when passing pointers data is serialised

A bio test case Gillespie simulator





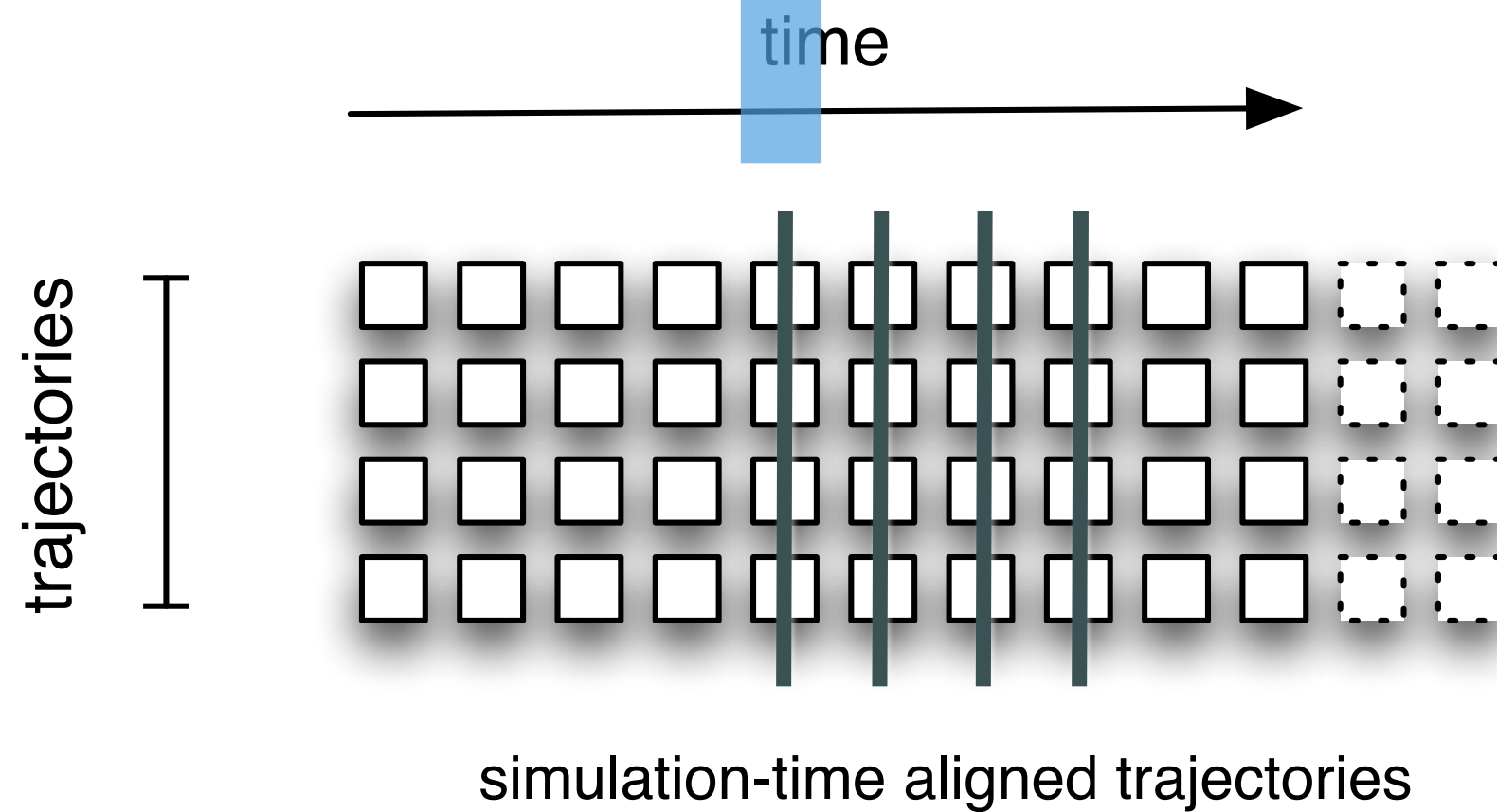
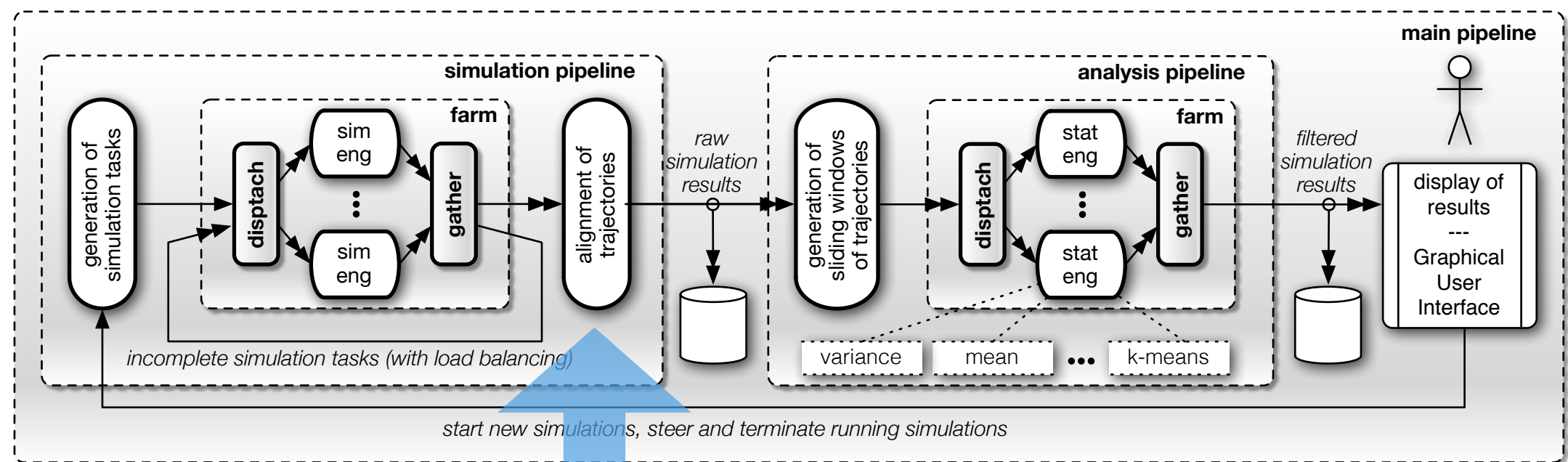
trajectories

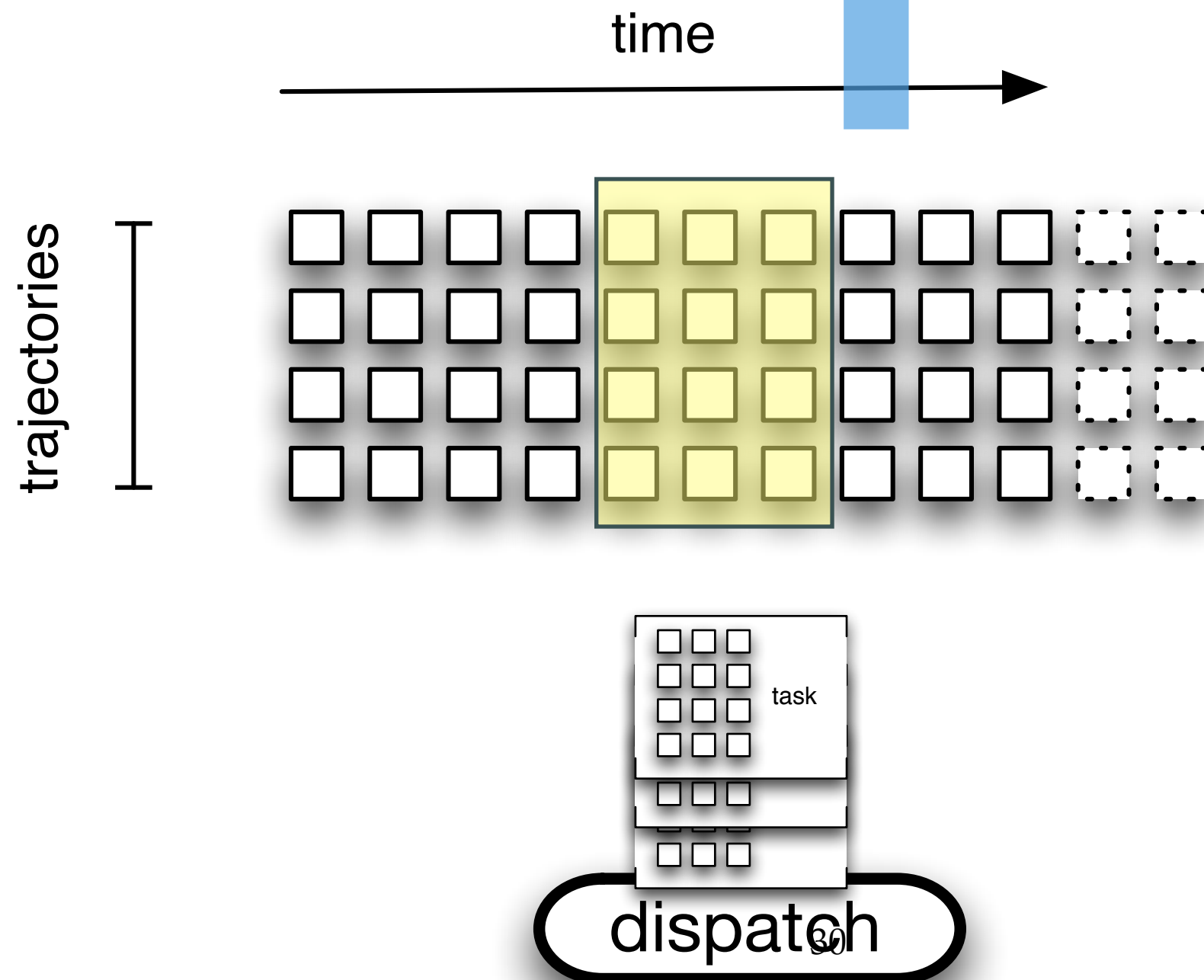
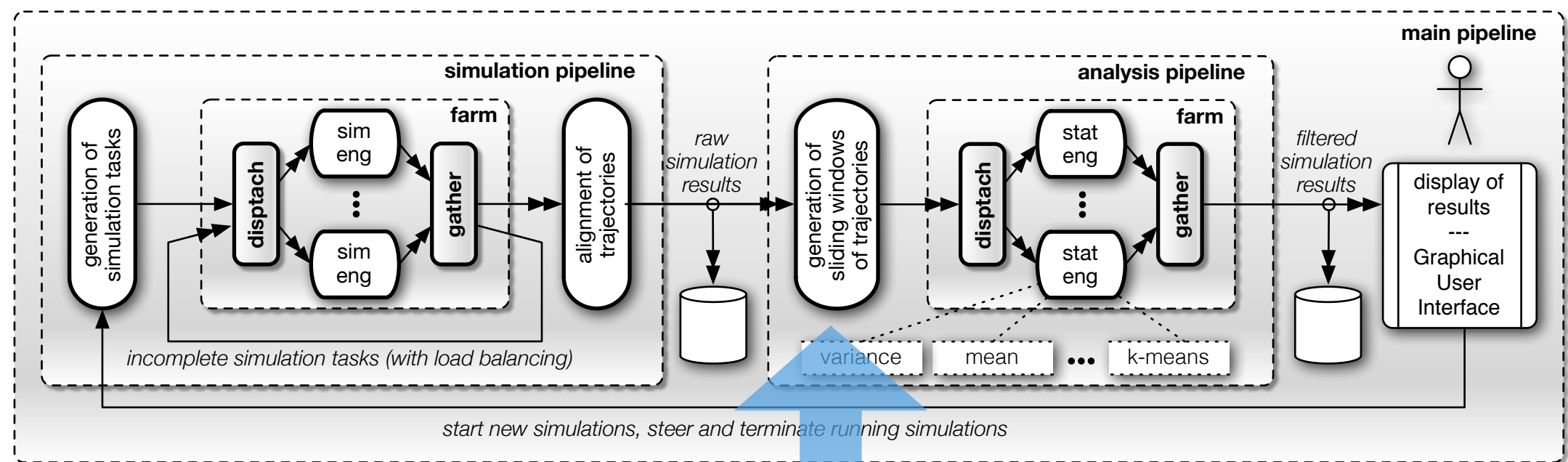


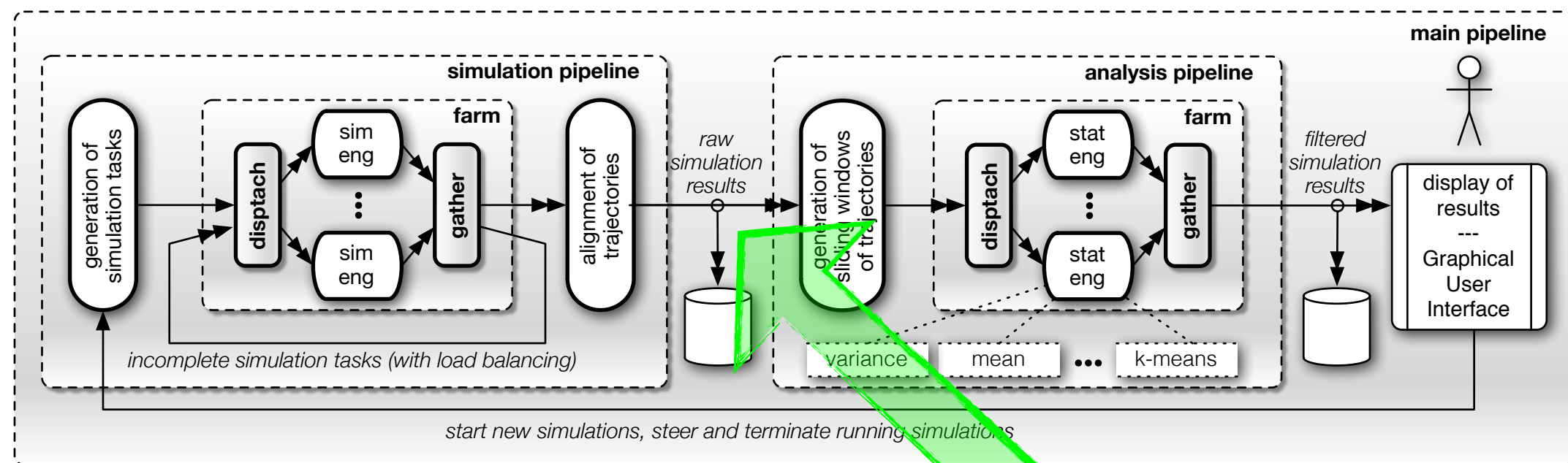
Trajectory 1

...

Trajectory n







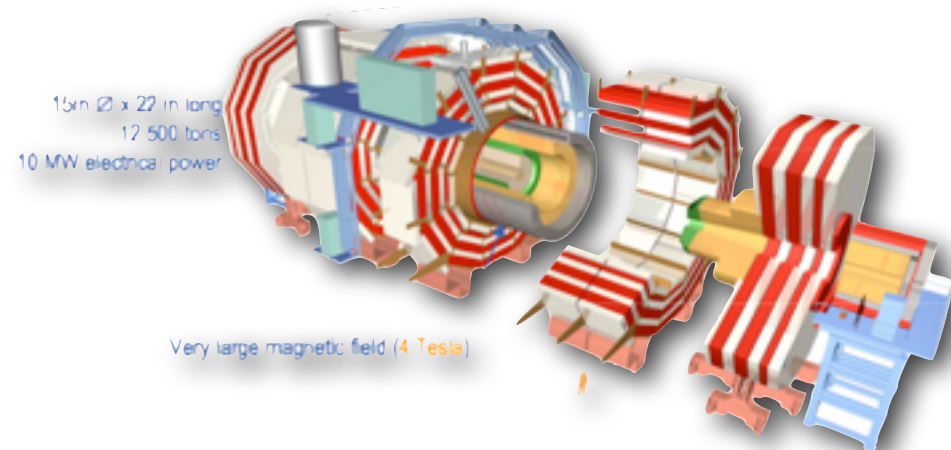
	Simulated time		Execution, general data			Seq	Parallel (16 core Intel SandyBridge)		
	time	resolution	raw data size	output size	MonteCarlo step latency	total time	total time	Throughput	speedup
Neurospora	1 month	~ 25 s	~8 GB	~6.5MB	600 ns	20 min	93 s	~20 MB/s	~16
Neurospora	4 days	~1 s	~80 GB	~65MB	1600 ns	60 min	~5 min	~280 MB/s	~16
Neurospora	1 month	~1 s	640 GB	520 MB	1600 ns	8 hours	~30 min	~280 MB/s	~16

what you need to store
and re-read with off-line filtering

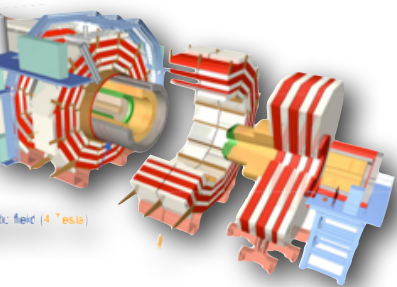
filtered data



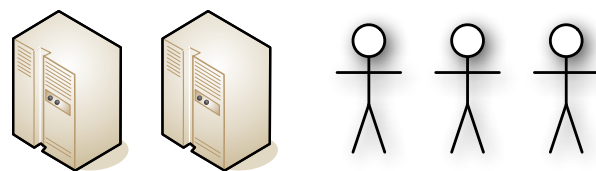
Innovative Methods for Particle Colliders at the Terascale (2012-2015)

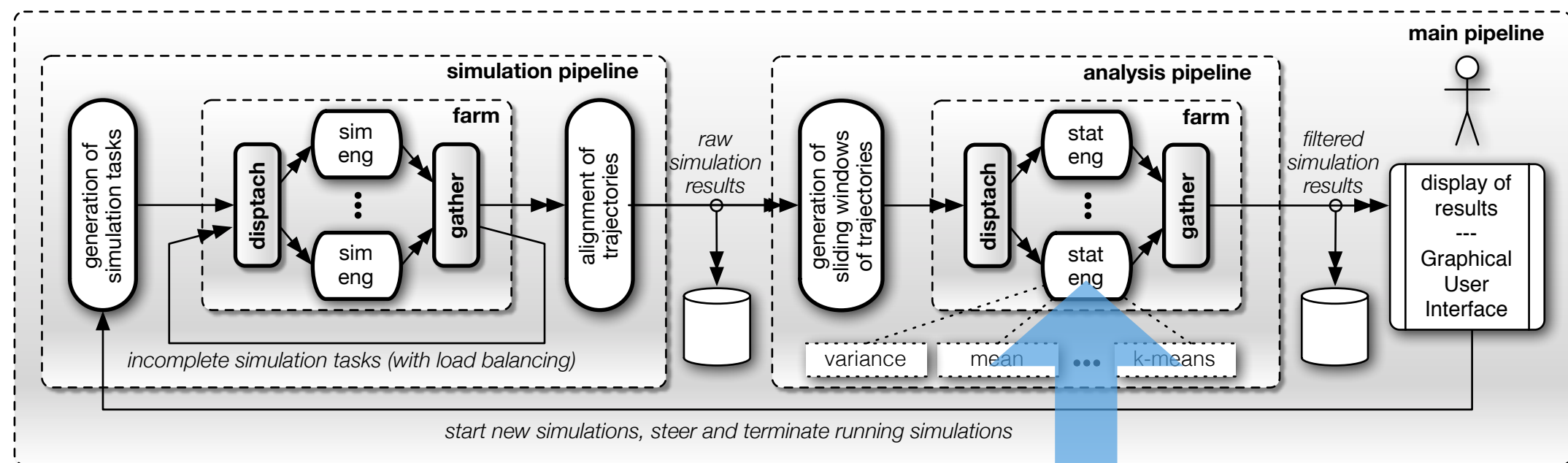


- CMS: Compact Muon Solenoid at CERN
 - 3500 scientists, 180 Universities and Research Labs (40 countries)
 - CMS is like a ~75 MegaPixels Digital Camera. 40M “photos” /s
Selection of 300 ‘photos’ /s ~**450 MB/s** from the detector are ~PBs of data /year



IMPACT project
contributed to results
on channel $H \rightarrow ZZ \rightarrow 4l$
shown at Higg's boson
claim Jul 2012



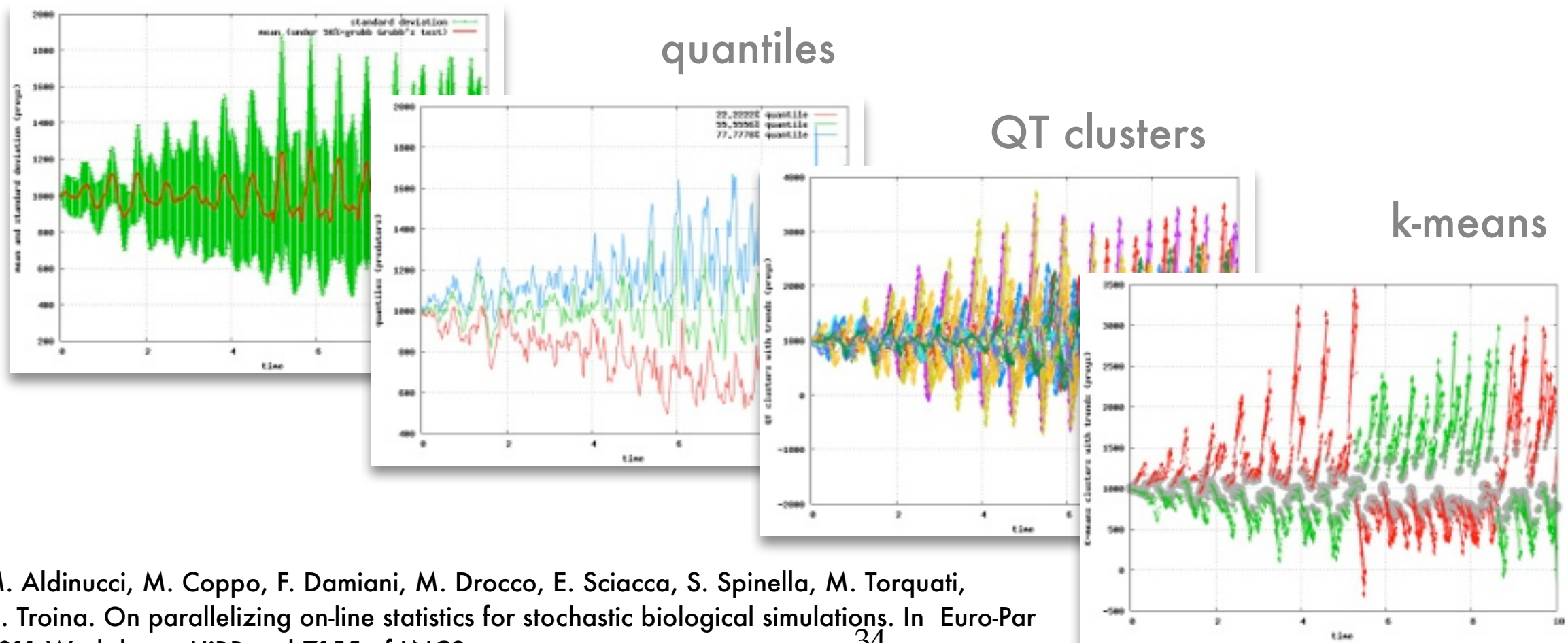


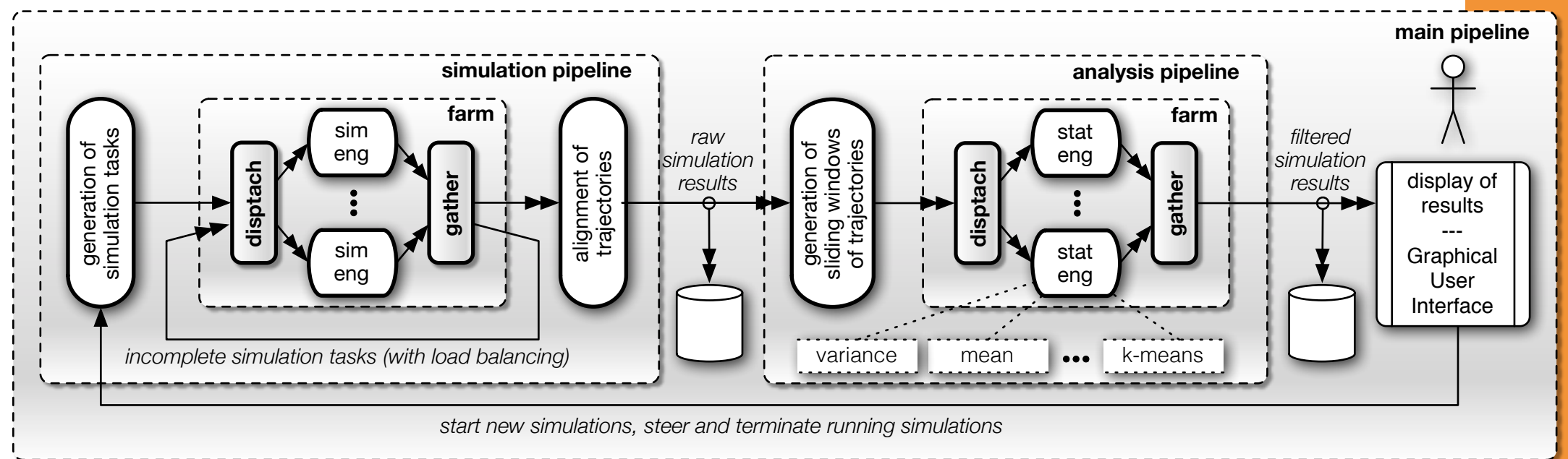
mean variance

quantiles

QT clusters

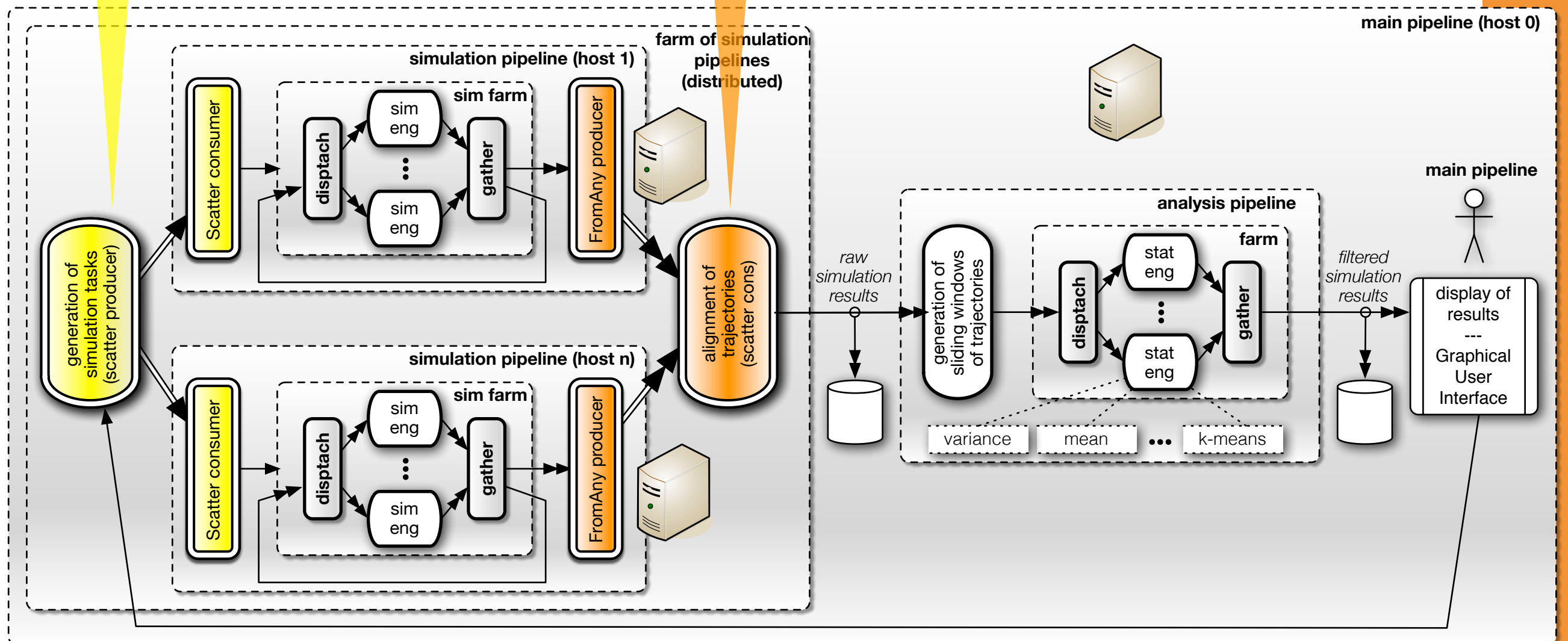
k-means



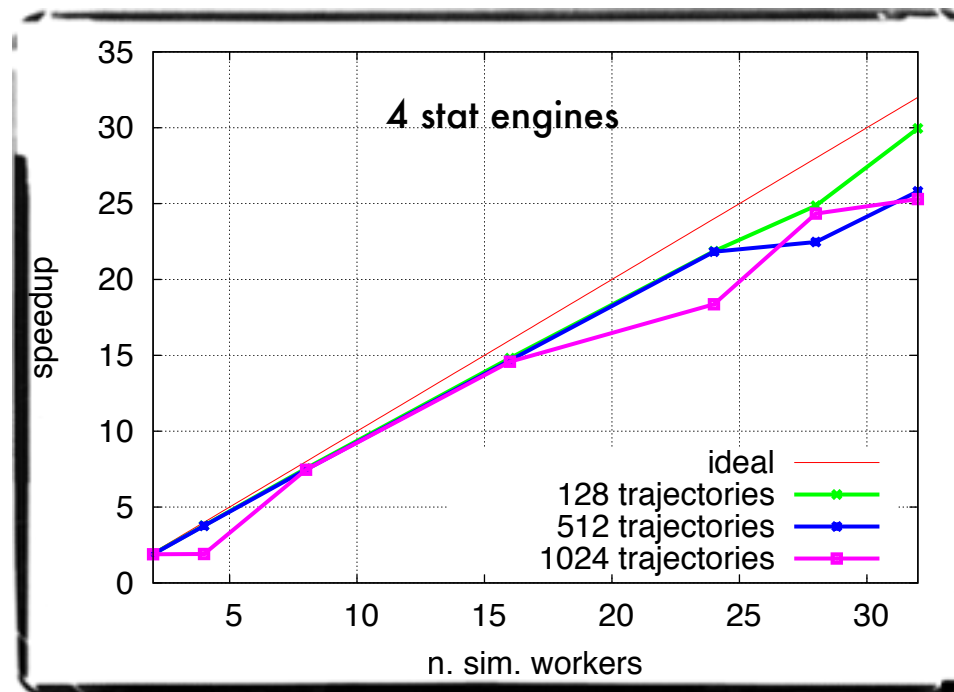
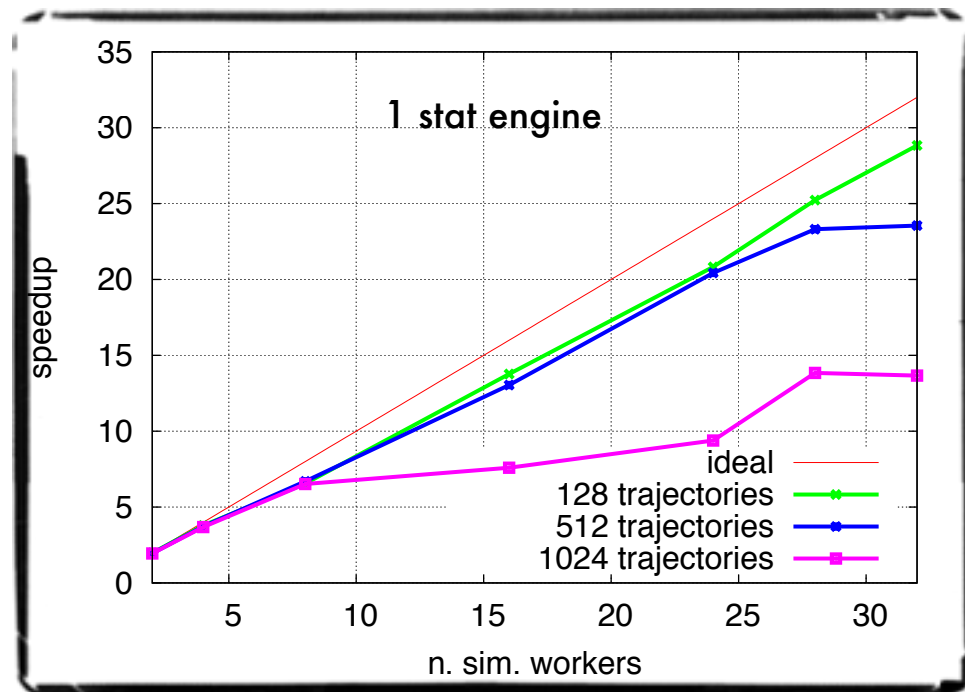


scatter seeds

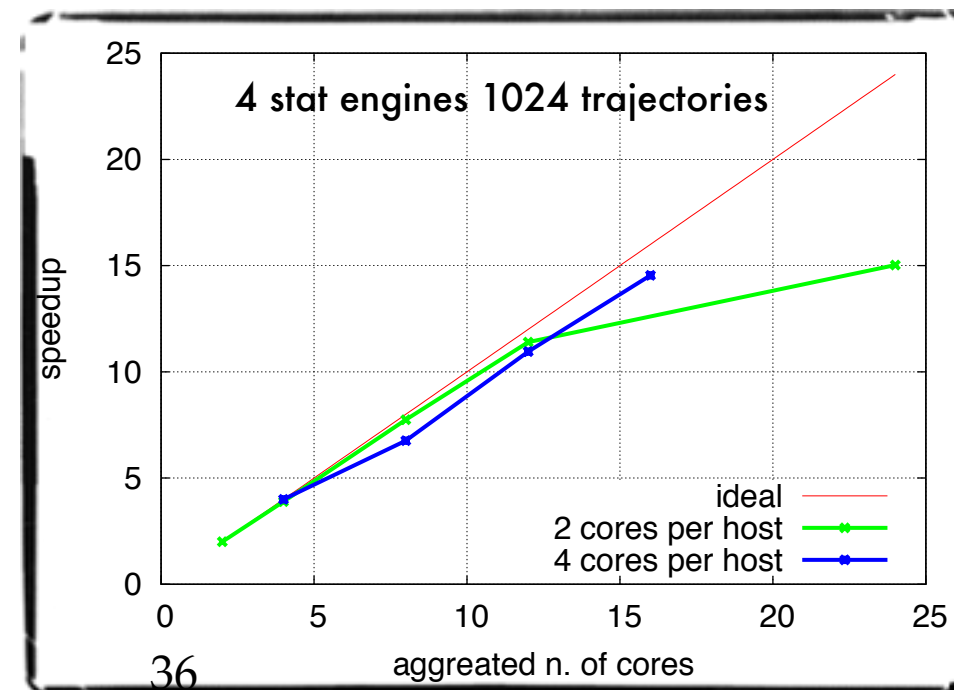
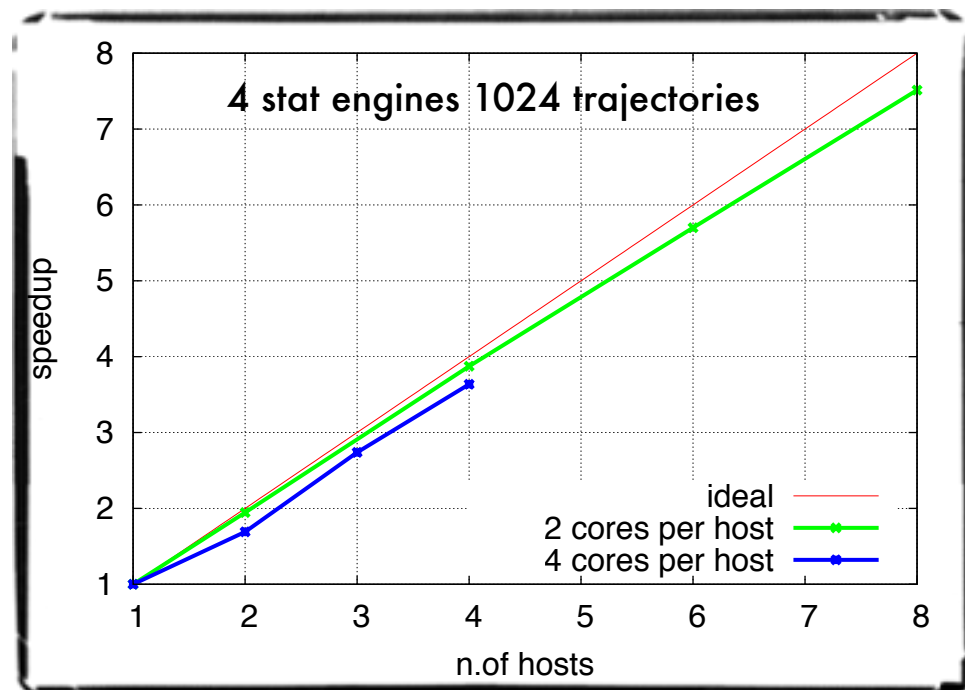
stream join (fromany)



Performance



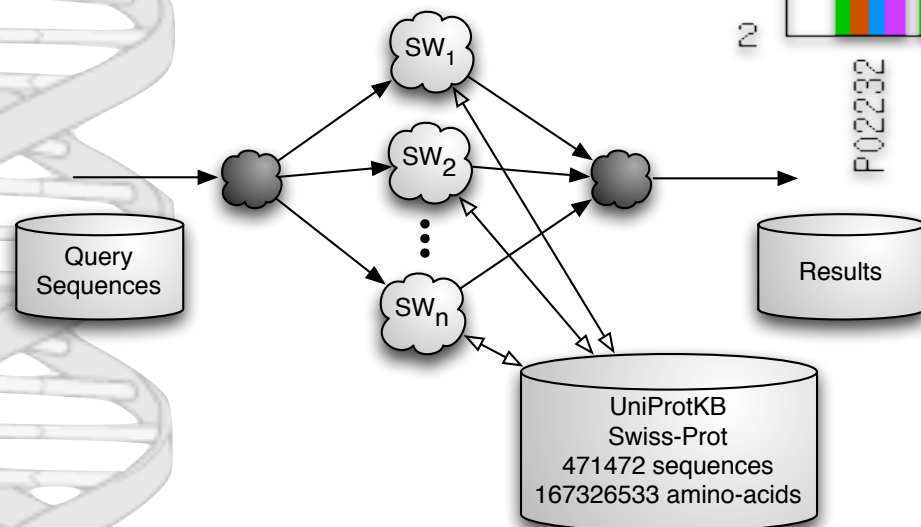
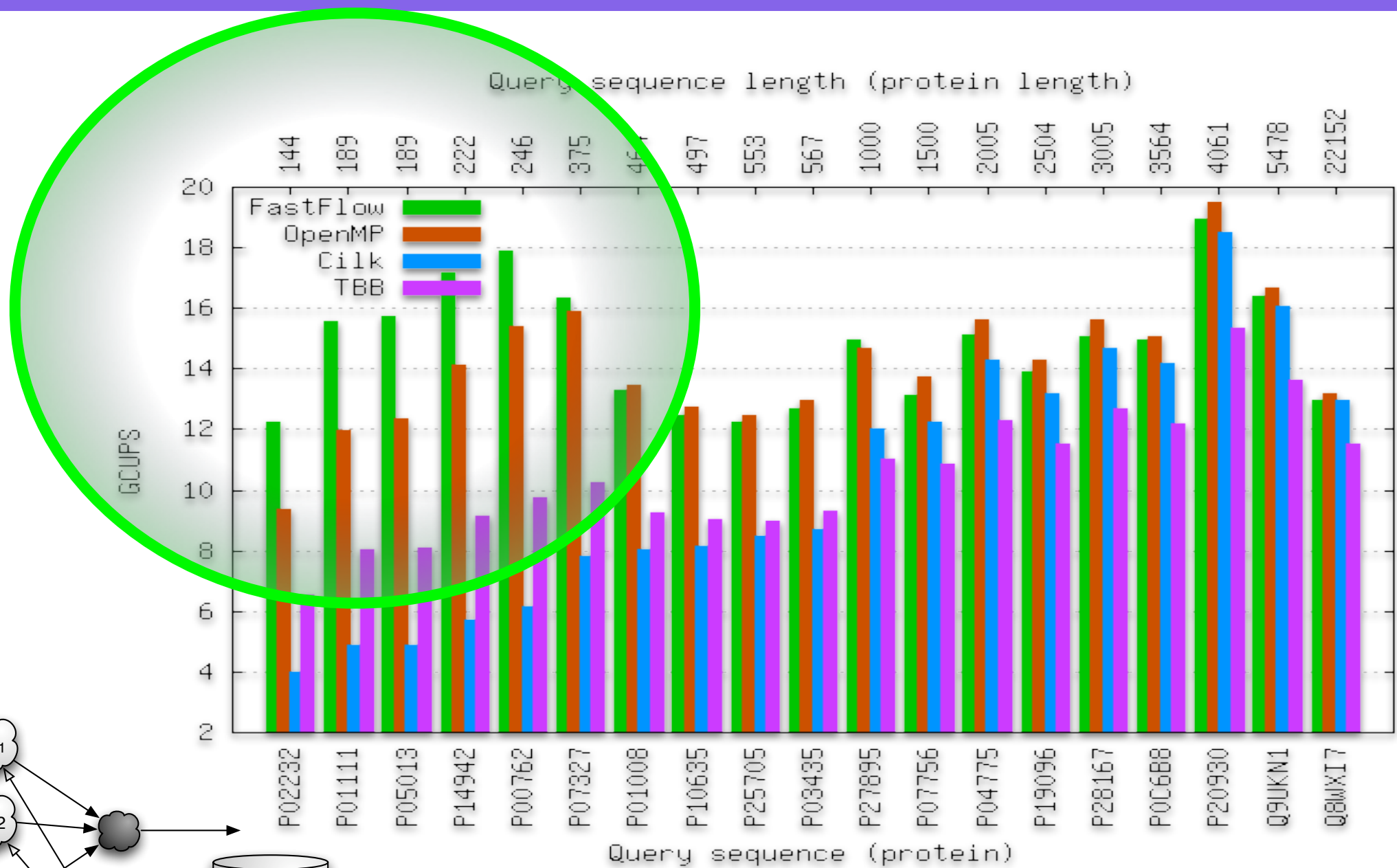
Intel
Nehalem
32-core



Cluster 8x
Intel Xeon 6-core
Infinband (IPoIB)

courtesy of
Mellanox

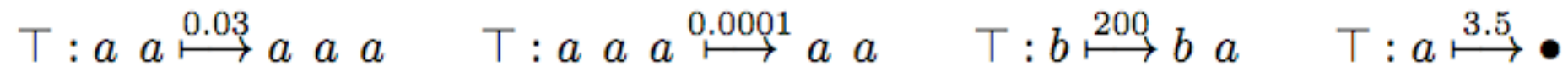
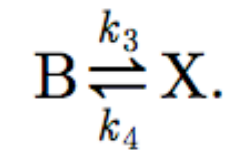
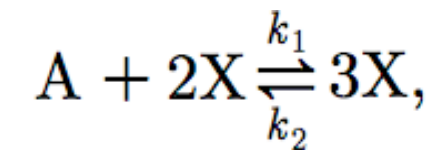
FastFlow-SWPS3 - Smith-Waterman (probably among the fastest SW implementation around)



Bio - some results

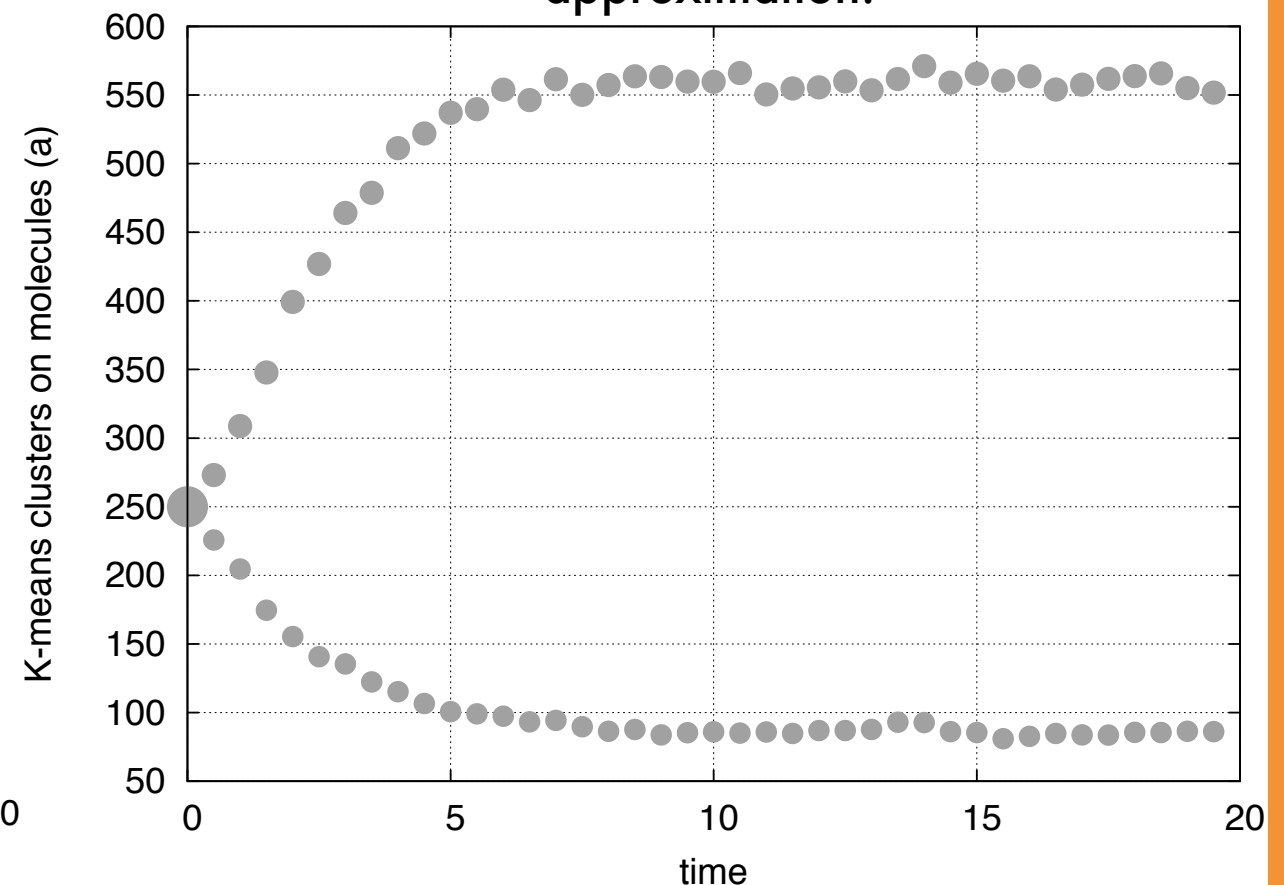
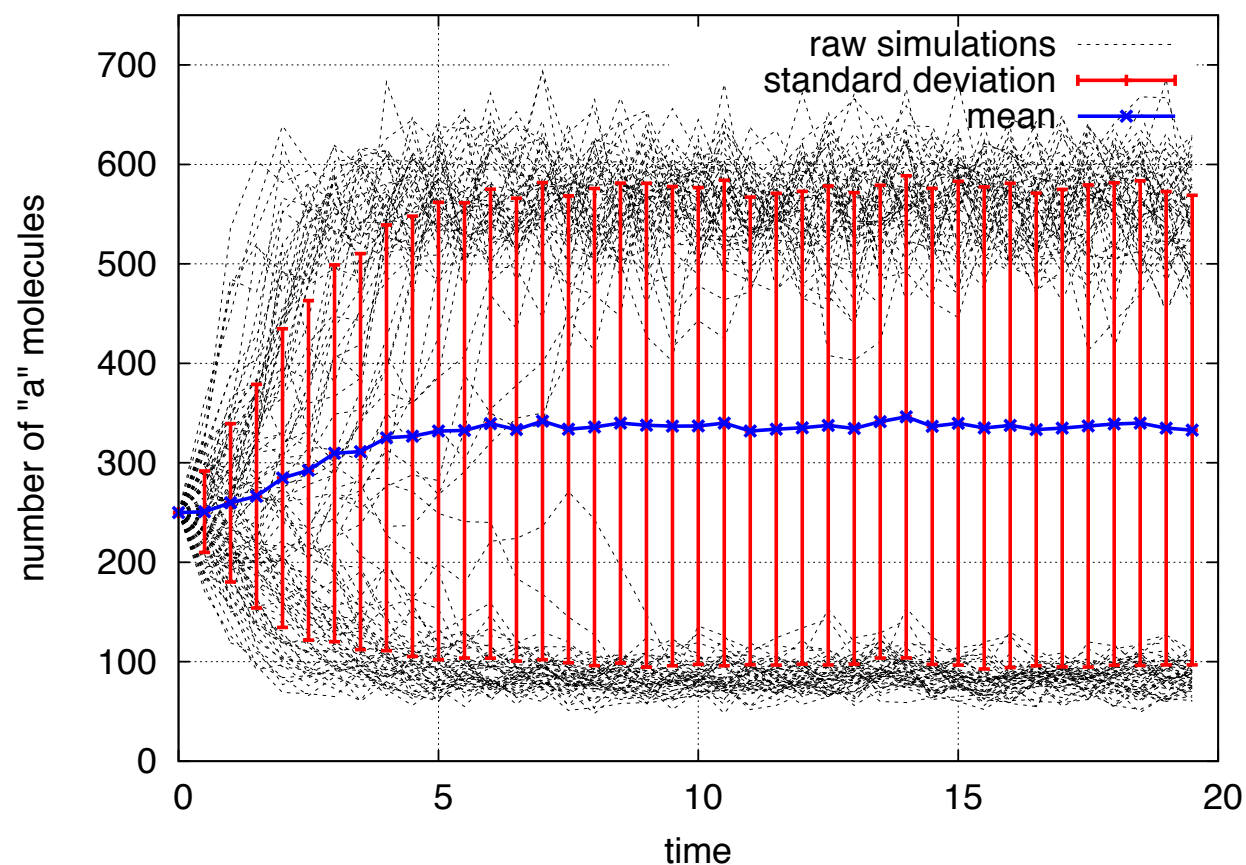
Schlögl model

autocatalytic, tri-molecular reaction
scheme (bistable)

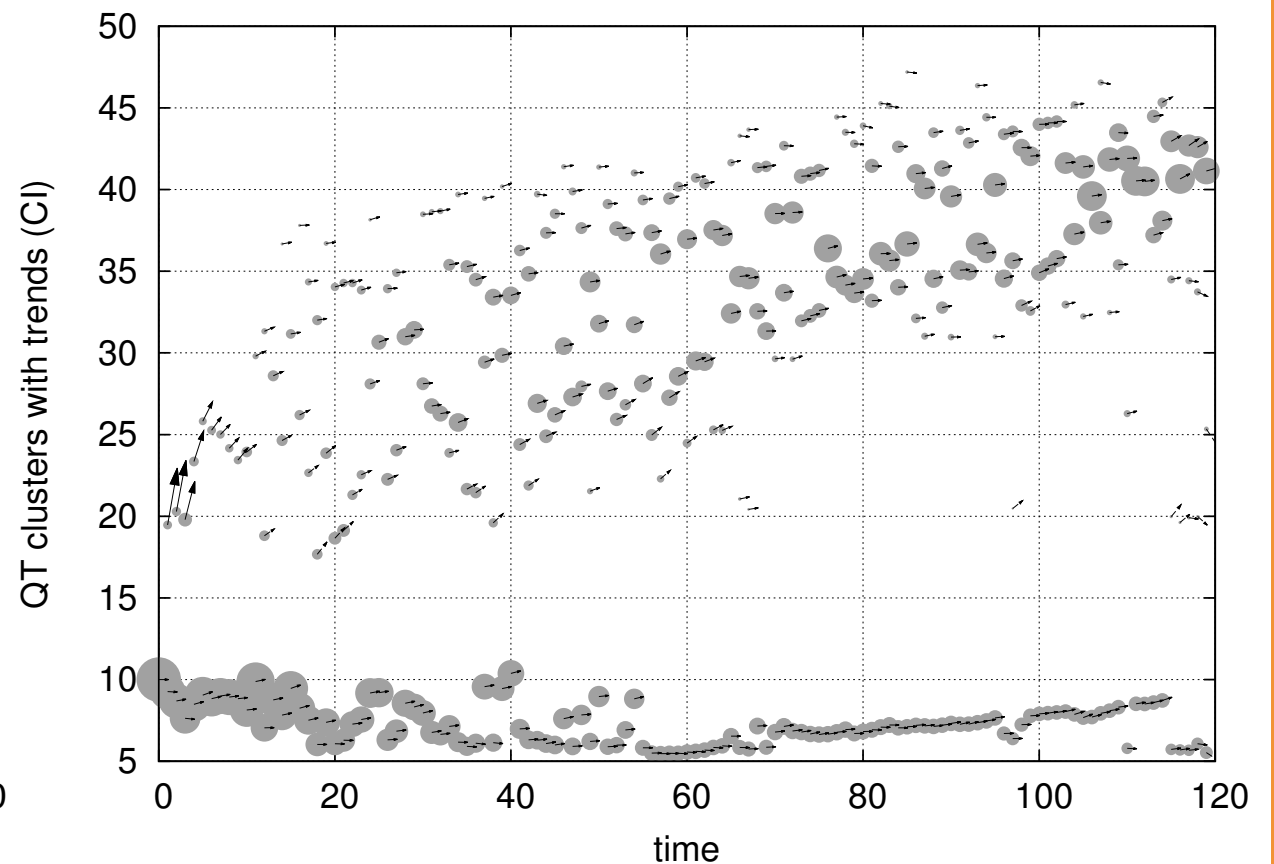
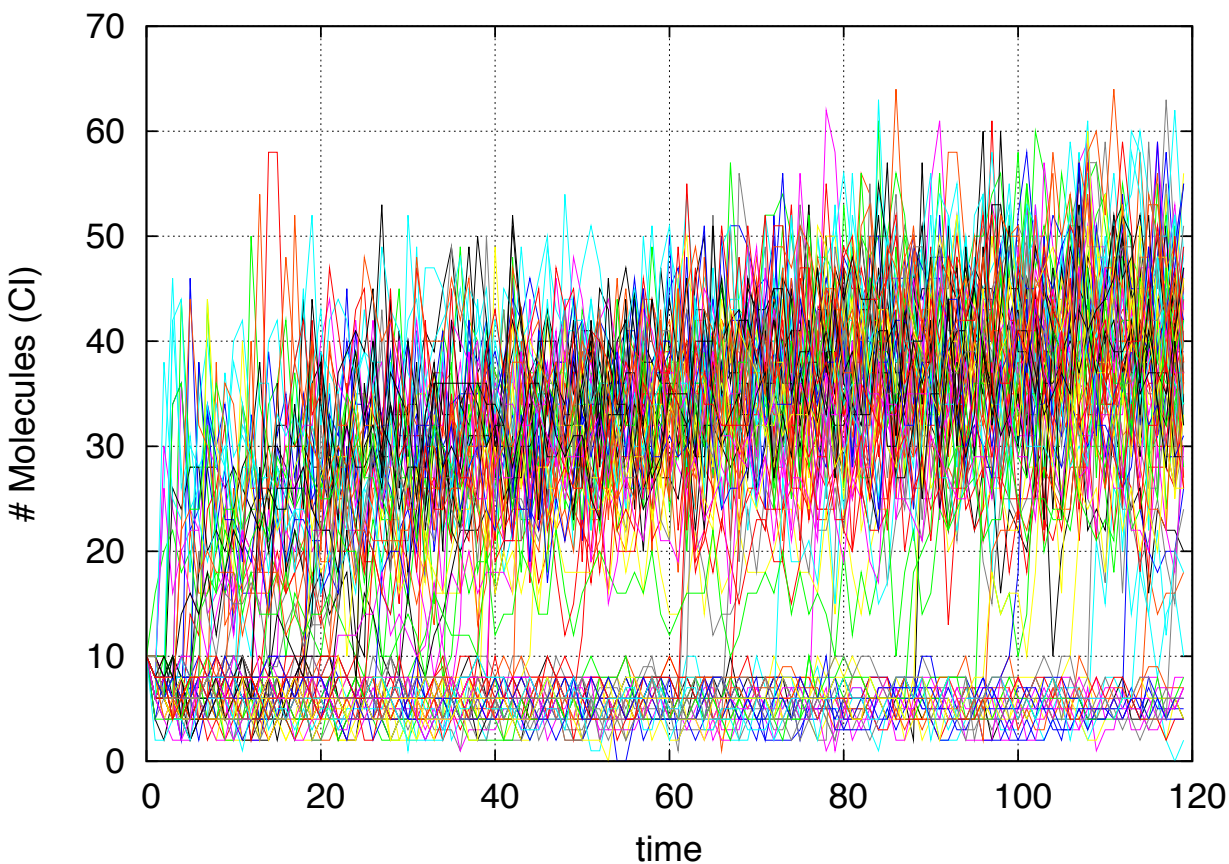
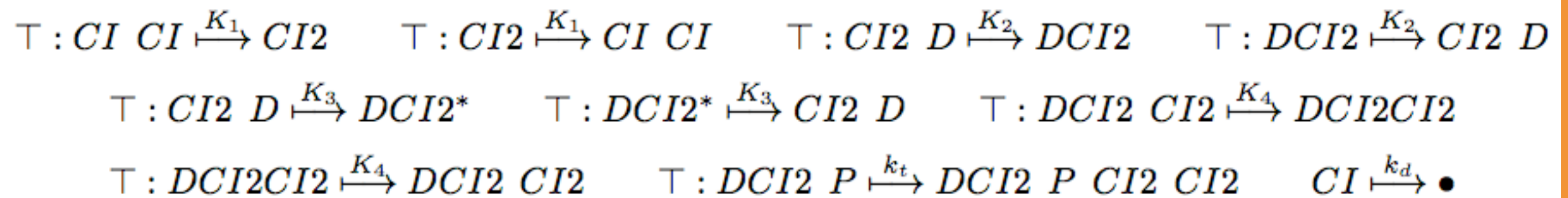
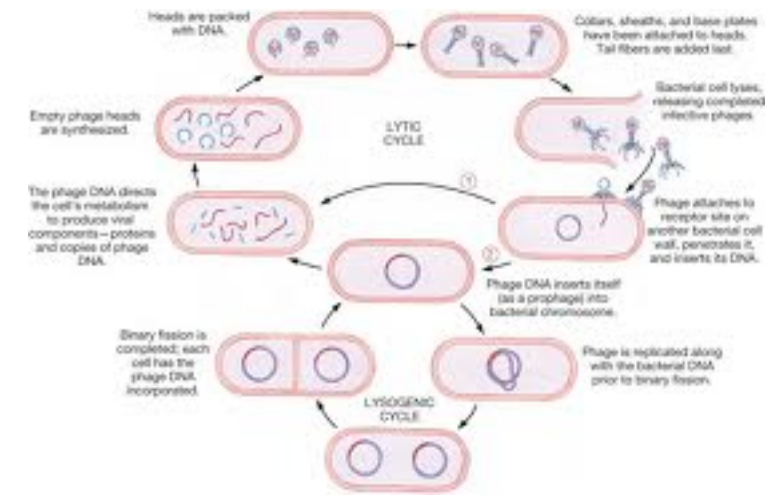


Notice this is a clustering of curves and is done
while the curves are not yet fully produced.

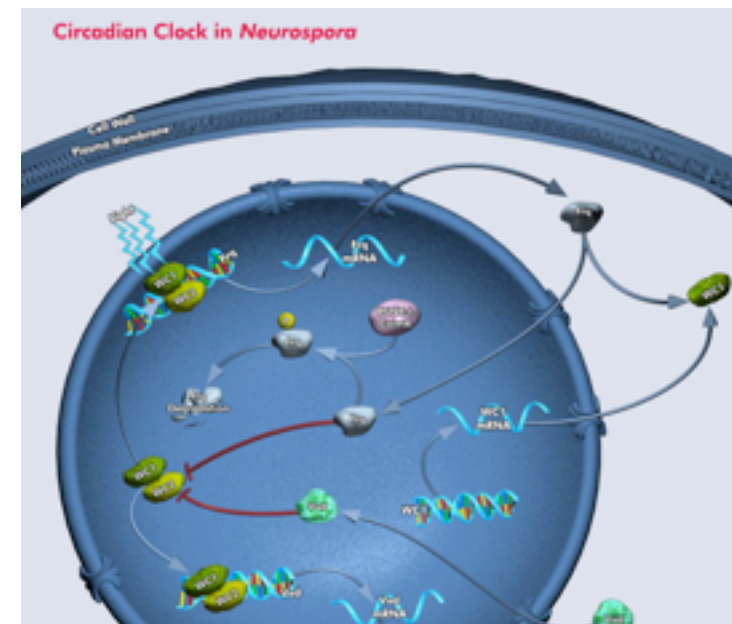
It can be done on-line with very good
approximation.



Bacteriophage λ life cycle integration of a strand of DNA in the molecule of E. coli DNA (multi-stable)



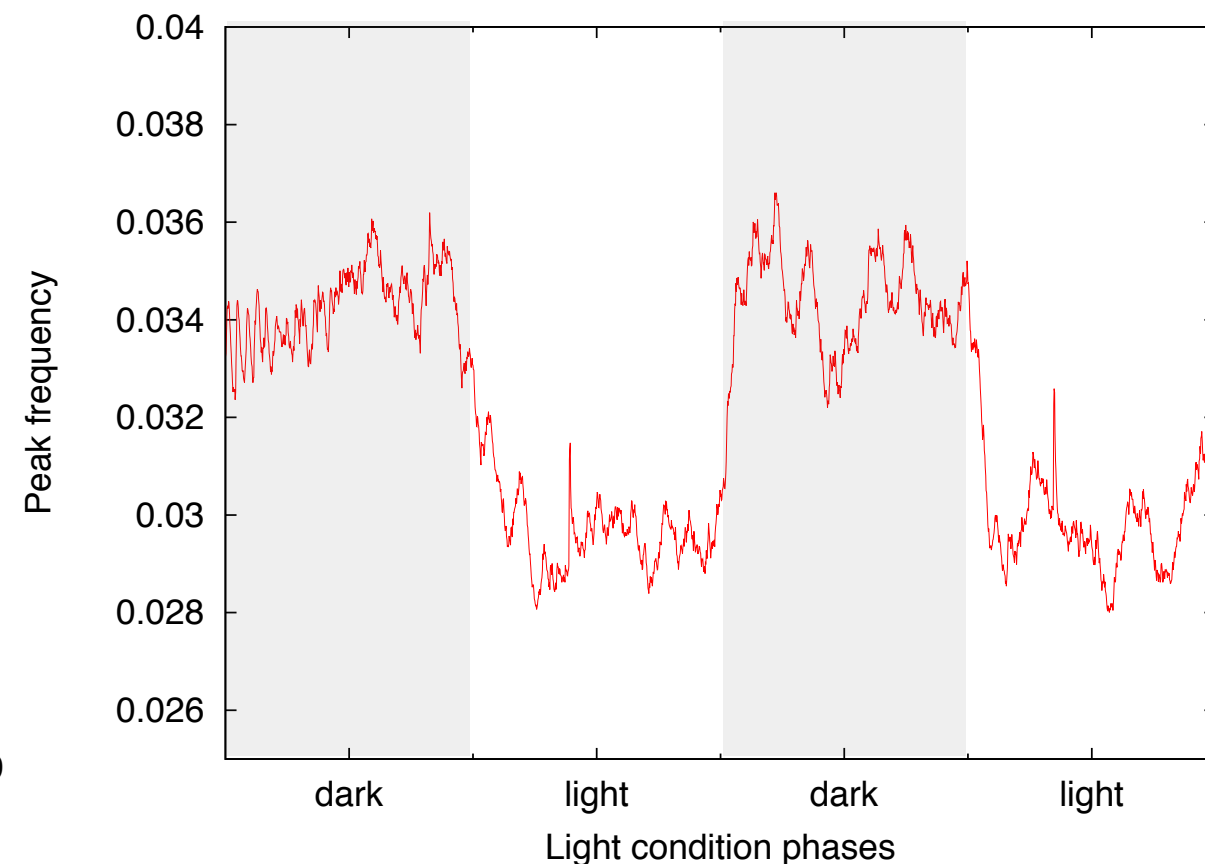
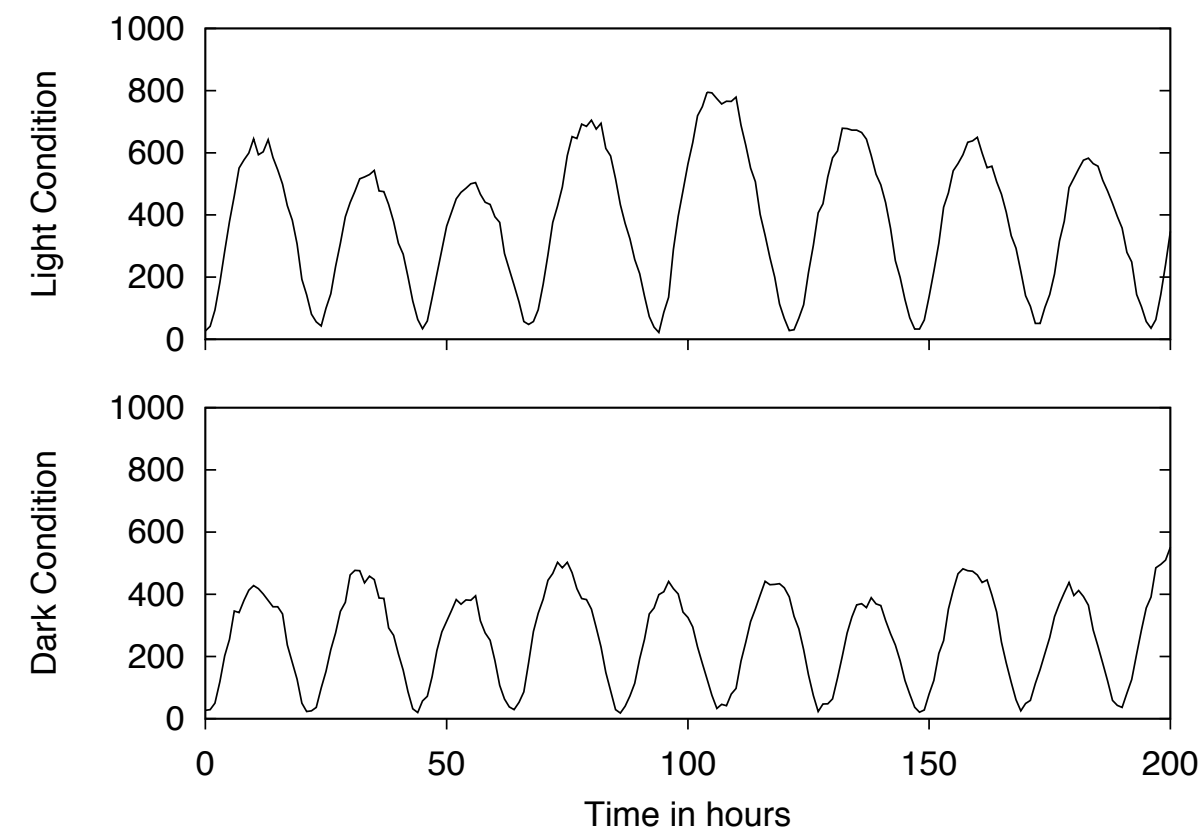
Transcriptional regulation in Neurospora (circadian clock period detection)



$$\top : (x \sqcup FRQ \ X)^\eta \xrightarrow{f_{FRQ}^{(t)}} (x \sqcup FRQ \ X)^\eta \ M \quad \top : M \xrightarrow{k_s} M \ FRQ$$

$$\top : M \xrightarrow{f_M} \bullet \qquad \top : FRQ \xrightarrow{f_d} \bullet$$

$$\top : FRQ(x \sqcup X)^\eta \xrightarrow{k_1} (x \sqcup FRQ X)^\eta \quad \top : (x \sqcup FRQ X)^\eta \xrightarrow{k_2} FRQ(x \sqcup X)^\eta$$



Conclusions

- Talk focused on programming model
 - streaming in particular
 - regards how build workflow for very high frequency
 - systems biology, physics, stock market trading, real-time processes, sensor stream-fusion, video / audio ...
 - data movement & high-level are key features
 - helps the mapping of features to platforms and performance portability
 - ease the design
 - **Patterns: MapReduce is only an example, it is not the only one!**
- Cloud?
 - yes, to make data & computation accessible, e.g. “analysis as a service”.
 - to build HPC workflows? Not sure about that, likely not.
- Formal biology probably at embryonal stage
 - similar data & computation problems in analysis “in vitro” experiments
 - garnering interest from industrial and “core” bio scientist for parallel computing

Acknowledgements

- Projects
 - Paraphrase: Parallel Patterns for Adaptive Heterogeneous Multicore Systems (EU-FP7 STREP, 2011-2014, 3.5 M€)
 - IMPACT: Innovative Methods for Particle Colliders at the Terascale (2012-2015, 400 K€)
 - HiPEAC: High Performance and Embedded Architecture and Compilation (EU Network of Excellence, 2012-2016)
 - BETTY: Behavioral Types for Reliable Large-Scale Software Systems (EU Cost Action, 2012-2016)
 - BioBITS: Converging technologies: Biotechnology-ICT (2008-2012)
- Partners: IBM, Mellanox
- Contributions for HPC from
 - Massimo Torquati (Pisa), Massimiliano Meneghin (IBM Research), Marco Danelutto (Pisa), Peter Kilpatrick (Belfast), Maurizio Drocco (Torino), Fabio Tordini (Torino)
- Contributions for biological models from
 - Luca Cardelli (Microsoft), Pietro Liò (Cambridge), Andrea Bracciali (Stirling), Eva Sciacca, Salvatore Spinella, Mario Coppo, Angelo Troina, Ferruccio Damiani, Cristina Calcagno (Torino)

